

版权注意事项：

- 1、书籍版权归作者和出版社所有
- 2、本PDF仅限用于个人获取知识，进行私底下的知识交流
- 3、PDF获得者不得在互联网上以任何目的进行传播
- 4、如觉得书籍内容很赞，请购买正版实体书，支持作者
- 5、请于下载PDF后24小时内删除本PDF。

学习量化金融领域强有力的工具，用R语言解决多种多样的问题

量化金融R语言 初级教程

Introduction to R for
Quantitative Finance

[匈牙利] Gergely Daróczy 等 著
高蓉 李茂 译



量化金融R语言 初级教程

[匈牙利] Gergely Daróczi 等 著
高蓉 李茂 译

人民邮电出版社

北 京

图书在版编目 (CIP) 数据

量化金融R语言初级教程 / (匈) 盖尔盖伊
(Gergely Daróczy) 等著 ; 高蓉, 李茂译. -- 北京 :
人民邮电出版社, 2017.5
ISBN 978-7-115-45123-1

I. ①量… II. ①盖… ②高… ③李… III. ①程序语
言—程序设计—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2017)第061199号

版权声明

Copyright ©2013 Packt Publishing. First published in the English language under the title *Introduction to R for Quantitative Finance*.

All rights reserved.

本书由英国 Packt Publishing 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

-
- ◆ 著 [匈牙利] Gergely Daróczy 等
译 高 蓉 李 茂
责任编辑 胡俊英
责任印制 焦志炜
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 10.5
字数: 190 千字 2017 年 5 月第 1 版
印数: 1—2 500 册 2017 年 5 月河北第 1 次印刷
- 著作权合同登记号 图字: 01-2016-5919 号
-

定价: 49.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第 8052 号

内容提要

R 是用于统计分析、绘图的语言和操作环境。它是属于 GNU 系统的一个自由、免费、源代码开放的软件，是一个用于统计计算和统计制图的优秀工具。

本书通过 9 章的内容向读者详细介绍使用 R 语言实现量化金融的一些基础知识和方法，内容包括时间序列分析、投资组合优化、资产定价模型、固定收益证券、估计利率期限结构、衍生品定价、信用风险管理、极值理论和金融网络等。

本书的目标读者是那些希望通过 R 语言来解决量化金融问题的读者，如果读者具备一定的金融知识，将会对本书的阅读有较大的帮助。通过阅读本书，读者将学习到有关 R 语言的诸多核心内容，并了解 R 语言在量化金融方面的各类应用。

由于量化金融技术的学习和应用难度，目前在世界范围内被广泛地采用，在中国也拥有大量用户。但是，可以结合量化金融与数据科学的图书，在市场上依然很少，而本书正是基于这种需求而作的关于量化金融的学习教程。本书选择 R 语言作为工具，通过 R 语言学习量化金融，帮助读者快速掌握知识并加以实践应用，为各种金融问题提供实践解决方案，同时还可使用第三方贡献的免费 R 包。

本书内容广泛，取舍兼得，涵盖了实证金融、金融工程、交易策略与银行管理等内容，可以帮助读者全面学习金融理论与技术。本书既包含金融时间序列分析、资产定价和期权定价等理论知识，还包括投资组合管理、信用风险管理等实践

译者序

自 20 世纪 50 年代马可维茨开创投资组合理论以来，金融学开始与数学模型和数据分析紧密融合，逐渐形成了高度数理化的现代金融学。时至今日，无论是对传统的金融学理论，还是对如日中天的量化投资，又或是对方兴未艾的金融科技（Fintech），数学模型与数据科学源源不断地为这些金融理论和实践提供着发展工具，一起构建了复杂深奥、丰富多彩的量化金融世界。由于数学模型与数据分析本身的复杂性，量化金融技术一直被公众视为数学家或华尔街的“火箭科学家”的秘密武器。

近年来，开源软件得到了迅速推广，促进了各行各业对数据科学的学习与应用。R 就是这样一种开源数据分析工具。它灵活易用，数据分析能力强大，从技术上降低了量化金融技术的学习和应用难度，目前在世界范围内被广泛地采用，在中国也拥有大量用户。但是，可以结合量化金融与数据科学的图书，在市场上依然很少，而本套书正是基于这种需求而作的关于量化金融的学习教程。本套书选择 R 语言作为工具，通过 R 语言学习量化金融，帮助读者快速掌握知识并加以实践应用，为各种金融问题提供实践解决方案，同时还可使用第三方贡献的免费 R 包。

本套书内容广泛、取舍精当，涵盖了实证金融、金融工程、交易策略与银行管理等内容，可以帮助读者全面学习金融理论与技术。本套书既包含金融时间序列分析、资产定价和期权定价等理论知识，也包括投资组合管理、信用风险管理等实践

知识,还涉及金融学的前沿领域以及 2008 年金融危机后发展起来的金融网络分析理论。本套书包括两本,分别是《量化金融 R 语言初级教程》和《量化金融 R 语言高级教程》。学习《量化金融 R 语言初级教程》不需要读者精通 R 和金融理论,但需要对金融领域具有一定了解。学习《量化金融 R 语言高级教程》要求读者具备一定的 R 编程能力和金融学基本概念,建议在学习《量化金融 R 语言初级教程》的基础上再学习《量化金融 R 语言高级教程》。

作为金融学教师,我们认为本套书是通向量化金融世界的一把“金钥匙”,可以有效促进金融知识和计算技术的传播。我们很荣幸地承担了本书的翻译工作。在此特别感谢胡俊英编辑,胡编辑认真专业的审核工作,有力保证了本书翻译的如期完成,我们亦受益匪浅。同时特别感谢天津理工大学的李茂老师,书中多处的专业词汇敲定,都来自与李茂老师的讨论结果。当然,文责自负。同时,感谢杭州电子科技大学 2016 年高等教育研究资助项目 YB201631“投资学教学与 R 软件应用”的支持。

——高蓉,2017 年 3 月

译者简介

高蓉，任教于杭州电子科技大学经济学院。博士毕业于南开大学经济学院，本科毕业于南开大学数学科学学院。研究领域包括资产定价、实证金融、数据科学应用。已出版教材《实验投资学》，译著《数据科学入门》，发表学术论文数篇。

李茂，任教于天津理工大学数学系，毕业于北京师范大学。热爱数据科学，从事与统计和数据分析相关的教学和研究工作。

“我非常感谢家庭成员的支持和理解。尽管为了完成本书的工作，他们已经很久没有见到我，但他们也依然默默支持着我。我还衷心感谢所有在匈牙利布达佩斯考文纽斯大学任教的诸位合作者，他们为本次合作提供了有用的内容。”

Michael Puhle 在德国帕绍大学 (University of Passau) 获得了金融学博士学位。他曾在慕尼黑的全联资产管理公司 (Allianz Global Investors) 担任高级风险控制经理多年，后来在毕马威金融风险管理 (KPMG's Financial Risk Management) 部门中担任管理助理，在那里他就市场风险敞口为银行提供咨询。他还是斯普林格出版社 (Springer Publishing) 出版的《债券组合优化》(Bond Portfolio Optimization) 的作者之一。

Edina Berlinger 是毕业于布达佩斯考文纽斯大学 (Corvinus University of

作者简介

Gergely Daróczy 是一位社会学博士学位候选人，拥有大约 8 年 R 编程的数据管理和分析任务的工作经验。**Gergely** 数年来在匈牙利的多所大学讲授统计学课程并从事数据分析工作，最近他还创建并协调着一个总部位于英国的在线报告创业公司。后者作为一种服务性平台，其软件或者平台称为 **rapporter.net**，可以对本书涉及的所有方法和技术提供一个直观的界面和接口。他对本书的贡献是提供了量化金融问题和方法的 R 实现。

“我非常感谢家庭成员的支持和理解。尽管为了完成本书的工作，他们已经很久没有见到我，但他们也依然默默支持着我。我还衷心感谢所有在匈牙利布达佩斯考文纽斯大学任教的诸位合作者，他们为本次合作提供了有用的内容。”

Michael Puhle 在德国帕绍大学（University of Passau）获得了金融学博士学位。他曾在慕尼黑的安联资产管理公司（Allianz Global Investors）担任高级风险控制经理多年，后来在毕马威金融风险管理（KPMG's Financial Risk Management）部门中担任管理助理，在那里他就市场风险模型为银行提供咨询。他还是斯普林格出版社（Springer Publishing）出版的《债券组合优化》（Bond Portfolio Optimization）的作者之一。

Edina Berlinger 是毕业于布达佩斯考文纽斯大学（Corvinus University of

Budapest) 的经济学博士。她是一名助理教授, 讲授公司金融、投资学和金融风险。她还担任大学金融系的领导职务, 也是匈牙利科学院金融分委员会的主席。她的专业涉及学生贷款系统、风险管理, 最近又涉及了网络分析领域。在学生贷款设计、流动性管理、异质代理模型和系统风险方面, 她领导过一些研究项目。

Péter Csóka 是布达佩斯考文纽斯大学的助理教授, 同时也是匈牙利科学院经济与区域研究中心、博弈论研究组的研究员。2008 年, 他在马斯特里赫特大学 (Maastricht University) 获得了博士学位。他的研究主题包括风险管理、风险资本配置、博弈论、公司金融以及一般均衡理论。他目前致力于分析对系统风险和非流动性资产组合的风险贡献。他在《运筹研究的数学方法》(Mathematical Methods of Operational Research)、《欧洲运筹研究杂志》(European Journal of Operational Research)、《博弈与经济行为》(Games and Economic Behaviour) 以及《银行与金融杂志》(Journal of Banking and Finance) 发表过论文。他还是布达佩斯金融市场流动性年度会议组织委员会的主席。

Daniel Havran 是匈牙利科学院经济研究所、经济与区域研究中心的博士后研究员。他还在布达佩斯考文纽斯大学担任兼职的助理教授, 在那里, 他讲授公司金融 (本科和博士水平) 和信用风险管理 (硕士水平) 的课程。2011 年, 他在布达佩斯考文纽斯大学获得了经济学博士学位。他研究的兴趣方向是公司现金、基金流动性管理以及场外市场的信用衍生品。

Márton Michaletzky 在 2011 年从布达佩斯考文纽斯大学获得了经济学博士学位。在 2000~2003 年间, 他是协和证券有限公司 (Concorde Securities Ltd) 的风险经理和宏观经济分析师。作为资本市场交易经理, 他在匈牙利国家高速公路管理公司获得了 30 亿欧元的证券化经验。2012 年, 他参与了一次 IPO 的准备工作以及匈牙利金融服务提供商的私人配售。在加入 DBH 投资之前, 他是 CUB 金融系的助理教授。

Zsolt Tulassay 在一家专业的美国投资银行任量化分析师, 从事评估衍生品定价模型相关的工作。在此之前, Zsolt 是布达佩斯考文纽斯大学金融系的助理讲师, 讲授衍生品、量化风险管理和金融计量经济学。Zsolt 拥有布达佩斯考文纽斯大学和中

欧大学的硕士学位。他研究的兴趣方向包括衍生品定价、构建收益率曲线、流动性风险以及异质代理模型。

Kata Vóradí 自 2013 年以来一直在布达佩斯考文纽斯大学任金融学的助理教授。2009 年, Kata 从布达佩斯考文纽斯大学研究生毕业, 并于 2012 年获得了博士学位, 她论文的主题是关于匈牙利股票市场的市场流动性风险分析。她的研究领域包括市场流动性、固定收益证券以及医疗系统的网络。除了研究, 她对教学也很积极。主要讲授公司金融、投资学、估值以及跨国金融管理。

Agnes Vidovics-Dancs 是博士学位候选人和布达佩斯考文纽斯大学金融系的助理教授。在此之前, 她是匈牙利政府债务管理局的初级风险经理。她的主要研究领域是通常的政府债务管理, 特别是主权危机和违约。

审稿人简介

Hari Shanker Gupta 博士是一位算法交易系统开发领域的高级量化研究分析师。在此之前，他是位于印度班加罗尔的印度科学研究院（Indian Institute of Science, IISc）的博士后。在 2010 年，他在 IISc 获得了应用数学和科学计算领域的博士学位，他还在印度瓦拉纳西的巴纳拉斯印度大学（Banaras Hindu University, BHU）完成了硕士学位。在瓦拉纳西，因为其杰出的表现，他获得过 4 次金奖。

Hari 已经在数学和科学计算的著名期刊上发表过 5 篇研究论文。他拥有在数学、统计和计算领域的工作经验，涉及数值方法、偏微分方程、数理金融、随机分析、数据分析、时间序列分析、有限差分以及有限元方法等领域。他擅长数学软件 Matlab、统计编程语言 R、编程语言 C，目前，他在 Python 平台下工作。

Ronald Hochreiter 是维也纳大学经贸大学的金融、会计和统计系的助理教授。在 2005 年，他在维也纳大学获得了他的计算管理科学的博士学位。他是一个热心的 R 用户，并开发了主要用于最优化建模和金融应用的 R 包。他的 R 项目汇总可以在 <http://www.hochreiter.net/R/> 找到，而且他的一些关于 R 使用的金融工程在线教程可以在 <http://www.finance-r.com/> 找到。

前言

本书将向你讲述如何使用统计计算语言 R 和量化金融知识来解决真实世界的量化金融问题。本书包括了丰富的主题，从时间序列分析到金融网络。每章都会简要地介绍理论知识并使用 R 来解决一个具体问题。

本书内容

第 1 章“时间序列分析”(Michael Puhle)，介绍了用 R 处理时间序列数据。并且，你会学到如何建模和预测房价，使用协整改善对冲比，以及对波动率建模。

第 2 章“投资组合优化”(Péter Csóka, Ferenc Illés, Gergely Daróczi)，包括了投资组合选择背后的理论思想，并说明了如何将这些知识运用于真实世界的的数据。

第 3 章“资产定价模型”(Kata Váradi, Barbara Mária Dömötör, Gergely Daróczi)，建立在前一章的基础上，给出了刻画资产收益率与风险之间关系的模型。本章包括资本资产定价模型和套利定价理论。

第 4 章“固定收益证券”(Márton Michaletzky, Gergely Daróczy), 是处理固定收益产品的基础。在这一章中, 你会学到如何计算这类产品的风险, 以及构建对利率变化免疫的投资组合。

第 5 章“估计利率期限结构”(Tamás Makara, Gergely Daróczy), 介绍了收益率曲线的概念, 并说明了如何使用政府债券的价格估计收益率曲线。

第 6 章“衍生品定价”(Ágnes Vidovics-Dancs, Gergely Daróczy), 使用离散和连续时间模型解释了衍生品定价。而且, 你还会学到如何计算衍生品风险的度量以及所谓的“希腊字母”。

第 7 章“信用风险管理”(Dániel Havran, Gergely Daróczy), 介绍了信用违约模型, 说明了如何使用 copula 对相关违约建模。

第 8 章“极值理论”(Zsolt Tulassay), 给出了极值理论在保险和金融中的可能应用。你将学到如何对火灾损失分布的尾部拟合模型。然后用拟合模型计算在险价值 (Value-at-Risk) 和预期损失值。

第 9 章“金融网络”(Edina Berlinger, Gergely Daróczy), 解释了金融网络在 R 中如何表示、模拟、可视化以及如何分析。我们将分析银行间借贷市场并学习如何系统化地检测重要的金融机构。

阅读本书之前的准备工作

本书提供的所有代码示例都应该在预装于计算机的 R 控制台上运行。你可以免费下载软件并找到安装所有主要操作系统的指导。尽管本书没有包括高级主题, 比如如何在整合发展环境中使用 R, 对于 Emacs、Eclipse、vi 或者 Notepad++ 以及其他的编辑器有许多很好的插件, 并且我们高度推荐你尝试 RStudio, 这是一款致力于 R 的免费开源的 IDE。

除了 R 的安装版本, 我们还会使用一些用户贡献的 R 包, 这些包可以很容易地从 CRAN (Comprehensive R Archive Network) 进行安装。要安装一个 R 包, 可以

在 R 控制台使用 `install.packages` 命令，如下：

```
> install.packages('zoo')
```

安装之后，这个包需要在使用之前载入到当前的 R 会话中：

```
> library(zoo)
```

在 R 的主页，你能找到免费的入门文章和手册，但本书面向初学者，因此并不需要读者具备额外的 R 语言知识。

目标读者

本书为那些希望使用 R 来解决量化金融问题的读者而写。我们假定读者对金融有一定了解，但本书也会介绍金融理论。我们并不需要读者熟悉 R，那些想开始学习 R 语言的读者会发现本书很有用，尽管我们没有给出完整的 R 语言概览，但说明了如何使用它的一部分来解决具体问题。即使你已经使用过 R，也会惊讶于它所能应用的问题广度。

排版约定

在本书中，你会发现多种文本样式，用以区别不同种类的信息。这里举例说明其中一些类型，及其含义的解释。

文本中的代码字、数据库表格的名字、文件夹名称、文件名称、文件扩展名、路径、虚拟 URL、用户输入以及推特的处理显示如下：“我们会运用一些 `forecast` 包中的方法。”

R 代码块（通常是个函数体）安排如下。

```
logreturn <- function(x) {  
  log(tail(x, -1) / head(x, -1))  
}
```

```
}
```

当我们希望一个代码块的特定部分能吸引你的注意，相应的行或者项会设置为**粗体**。

```
logreturn <- function(x) {  
  log(tail(x, -1) / head(x, -1))  
}
```

任何命令行的输入和输出的格式如下。

```
> pi  
[1] 3.141593
```

其中 R 控制台中显示的“>”表示等待处理的命令。多行表达式会在第一行显示相同的符号，但其余诸行会在开头有一个“+”号，表示后面的 R 表达式尚待完成。

新术语和重要词汇以黑体表示。你在屏幕上看到的文字，例如，在菜单中或在对话框中，就像这样出现在文本中：“按下 **Next** 键，就翻到下一屏。”



警告或者重要的注解出现在这样的图标中。



提示或者技巧出现在这样的图标中。

读者反馈

我们始终欢迎读者的反馈。如果你对本书有任何想法，喜欢或者不喜欢什么，请让我们知道。读者的反馈对我们来说非常重要，这样我们才能出版读者最需要的图书。

一般性的反馈，请通过电子邮件发送到 feedback@packtpub.com，请在邮件的主

题中注明书名。

如果你精通某个领域并有兴趣写书或参与写书，请参考我们的作者指南 www.packtpub.com/authors。

客户支持

现在，你已经是一位 Packt 图书的拥有者，我们会竭尽全力帮助你充分利用手中的书。

下载示例代码

可以使用你的账户从 <http://www.packtpub.com> 下载所有已购买的 Packt 图书的示例代码文件。如果你从其他地方购买了本书，可以访问 <http://www.packtpub.com/support> 并注册，我们会通过电子邮件把文件发送给你。

勘误表

虽然我们已经竭力确保本书内容正确，但疏漏之处在所难免。如果你在我国的图书中发现错误，无论是文本还是代码，希望能通知我们，我们将不胜感激。这样做可以减少其他读者的困扰，帮助我们改进本书的后续版本。如果你发现任何错误，请访问 <http://www.packtpub.com/submit-errata>，选择相应的图书，单击勘误表提交表单的链接，并输入详细的说明，然后提交。勘误一经核实，你的提交就被接受，此勘误将上传到本公司网站或者添加到现有勘误表。

要查看之前提交的勘误，登录 <https://www.packtpub.com/books/content/support> 并在搜索框中输入书名。请求的信息会在勘误部分出现。

侵权行为

互联网上的侵权材料是所有媒体都要面对的问题。在 Packt，我们非常重视保护版权和许可证。如果你发现我们的作品在互联网上有任何形式的非法拷贝，请立即为我们提供网址或者网站名称，以便我们能够寻求补救。

请把可疑盗版材料发送到 copyright@packtpub.com。

非常感谢你帮助我们保护作者以及宝贵的正版图书资源。

问题

如果你对本书内容有疑问，不管是哪个方面的，都可以通过 questions@packtpub.com 来联系我们，我们将尽最大努力解决。

第4章 固定收益证券	2.1	38
4.1 度量固定收益证券的市场风险	4.1.1	38
4.2 固定收益投资组合的免疫	4.2.1	62
4.2.1 净值免疫	4.2.2	62
4.2.2 目标收益免疫	4.2.3	63
4.3 可转换债券的定价	4.4	67

目录

第1章 时间序列分析	1
1.1 使用时间序列数据	1
1.2 对英国房屋价格建模并预测	5
1.2.1 模型识别和估计	6
1.2.2 模型诊断检查	7
1.2.3 预测	9
1.3 协整	9
1.4 波动率建模	13
1.4.1 风险管理的波动率预测	14
1.4.2 检验 ARCH 效应	14
1.4.3 GARCH 模型设定	16
1.4.4 GARCH 模型估计	16
1.4.5 回测风险模型	17
1.4.6 预测	20

1.5 小结	21
第 2 章 投资组合优化	22
2.1 均方差模型	24
2.2 解的概念	25
2.3 使用真实数据	27
2.4 切线组合和资本市场线	35
2.5 协方差矩阵中的噪声	36
2.6 如果方差不够用	37
2.7 小结	37
第 3 章 资产定价模型	39
3.1 资本资产定价模型	39
3.2 套利定价理论	41
3.3 贝塔估计	42
3.3.1 数据选择	43
3.3.2 简单贝塔估计	46
3.3.3 基于线性回归估计贝塔	46
3.4 模型检验	50
3.4.1 数据收集	50
3.4.2 对 SCL 建模	53
3.4.3 检验个体方差的解释能力	55
3.5 小结	57

第4章 固定收益证券	58
4.1 度量固定收益证券的市场风险	58
4.2 固定收益投资组合的免疫	62
4.2.1 净值免疫	62
4.2.2 目标日期免疫	63
4.2.3 定制	63
4.3 可转换债券的定价	63
4.4 小结	67
第5章 估计利率期限结构	68
5.1 利率期限结构与相关函数	68
5.2 估计问题	69
5.3 基于线性回归的期限结构估计	70
5.4 三次样条回归	71
5.5 R 函数应用	74
5.6 小结	77
第6章 衍生品定价	78
6.1 Black-Scholes 模型	78
6.2 Cox-Ross-Rubinstein 模型	81
6.3 两种模型之间的联系	84
6.4 希腊字母	86
6.5 隐含波动率	90

6.6 小结	91
第 7 章 信用风险管理	93
7.1 信用违约模型	94
7.1.1 结构模型	94
7.1.2 强度模型	100
7.2 相关违约——投资组合方法	102
7.3 迁移矩阵	104
7.4 使用 R 的信用评分入门	105
7.5 小结	106
第 8 章 极值理论	107
8.1 理论概览	108
8.2 应用——保险理赔的建模	109
8.2.1 探索性数据分析	110
8.2.2 理赔的尾部行为	111
8.2.3 阈值的决定	113
8.2.4 对尾部拟合 GPD 分布	114
8.2.5 使用拟合的 GPD 模型估计分位数	116
8.2.6 使用拟合的 GPD 模型计算预期损失	117
8.3 小结	118
第 9 章 金融网络	120
9.1 金融网络的表示、模拟和可视化	120

9.2 网络结构的分析和拓扑改变的检查	125
9.3 对系统风险的贡献——系统重要性金融机构的识别	131
9.4 小结	133
参考文献	135

第 1 章 时间序列分析

时间序列分析研究的是按时间顺序收集的数据。相邻的观测数据通常相互关联。因此，时间序列分析的技术需要处理这种相依性。

本章的目标是通过一些特定应用来介绍一些常用建模技术。我们将看到如何使用 R 来解决现实中的这些问题。首先，我们考虑如何在 R 中存储和处理时间序列。接着，我们处理线性时间序列分析，并展现如何将它用于建模和预测房屋价格。其次，我们通过考虑长期趋势，使用协整的概念来改进基本的最小方差对冲比。最后，本章讲述如何将波动率模型运用于风险管理。

1.1 使用时间序列数据

用于存储时间序列数据的基本 R 类有 `vector`、`matrix`、`data.frame` 以及 `ts` 对象。但是，它们可以存储在这些对象中的数据类型相当有限，并且，这些表达方式提供的方法范围也很有限。不过幸运的是，同名的包中的特定对象，`zoo`、`xts` 或 `timeSeries` 对象，对时间序列数据提供了更一般的表达形式。

对每个时间序列分析问题都新建时间序列对象是不必要的，但是复杂程度较高的分析则需要创建时间序列对象。你可以先将时间序列数据存储成向量形式，再计算数据的均值和方差。但如果你要用 `decompose` 对数据做季节分解，那就必须将

使用下面的命令，我们可以提取时间序列开头部分数据（美国和加拿大股票）

下面我们将使用 `tail` 函数来提取时间序列的末尾部分数据（日本和德国股票）

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

```
27.59 25.22 25.52 23.40
```

```
> tail(aapl)
```

```
2013-04-17 2013-04-18 2013-04-19 2013-04-22 2013-04-23 2013-04-24
```

```
402.80 392.82 390.53
```

第 1 章

时间序列分析

使用下面的命令，可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

使用下面的命令，我们可以找出苹果股价在所有时间中的高点，和这个高点发生的日期

1.1 使用时间序列数据

用于存储时间序列数据的基本 R 类有 `vector`、`matrix`、`data.frame` 以及 `ts` 对象。但是，它们可以存储在这些对象中的数据类型相当有限。并且，这些表达方式提供的方法范围也很有限。不过幸运的是，同名的包中的特定对象，`zoo`、`xts` 或 `timeSeries` 对象，对时间序列数据提供了更一般的表达形式。

对每个时间序列分析问题都创建时间序列对象是不必要的，但是复杂程度较高的分析则需要创建时间序列对象。你可以先将时间序列数据存储在向量形式，再计算数据的均值和方差，但如果你想用 `decompose` 对数据做季节分解，那就必须将

数据存储在时间序列对象中。

下面的例子假定你使用了 zoo 对象，因为 zoo 对象是使用最广泛的包之一。在使用 zoo 对象之前，需要使用下面的命令安装并载入 zoo 包（如果你已经安装，那只需要载入它）。

```
>install.packages("zoo")  
>library("zoo")
```

为了熟悉可用方法，我们使用苹果公司股票的日收盘价，创建一个名为 aapl 的 zoo 对象，存储在 CSV 文件 aapl.csv 中。表格的每一行包括一个日期和一个价格，两项通过逗号分隔。第一行包含了列名（Date 和 Close）。日期格式符合 ISO8601 推荐的基本标准符号（YYYY-MM-DD）。收盘价根据股票的拆分、股利以及相关改变进行调整。

小提示 下载示例代码



你对于在 <http://www.packtpub.com> 网站购买的所有 Packt 图书，都可以用自己在的账户从网站下载示例代码。如果你从其他途径购买了书籍，则可以访问 <http://www.packtpub.com/support> 并注册账号，示例代码会直接通过电子邮件发送给你。

使用下面的命令，可以从当前工作目录载入数据。

```
> aapl<-read.zoo("aapl.csv",  
+ sep="," , header = TRUE, format = "%Y-%m-%d")
```

为了初步了解数据，我们画出股票价格图形，并为整个图形设定一个标题（使用 main 参数）和对 x 轴和 y 轴标注了名称（分别使用 xlab 和 ylab）。

```
> plot(aapl, main = "APPLE Closing Prices on NASDAQ",  
+ ylab = "Price (USD)", xlab = "Date")
```

使用下面的命令，我们可以提取时间序列开头部分或结尾部分。

```
> head(aapl)
2000-01-03 2000-01-04 2000-01-05 2000-01-06 2000-01-07 2000-01-10
      27.58      25.25      25.62      23.40      24.51      24.08
> tail(aapl)
2013-04-17 2013-04-18 2013-04-19 2013-04-22 2013-04-23 2013-04-24
      402.80      392.05      390.53      398.67      406.13      405.46
```

使用下面的命令，可以找出苹果股价在所有时间中的高点，和这个高点发生的日期。

```
> aapl[which.max(aapl)]
2012-09-19
      694.86
```

当处理时间序列时，通常收益率更受关注，价格却不会。其原因是收益率通常平稳。因此我们会计算简单收益率或连续复合收益率（按百分比的形式）。

```
> ret_simple <- diff(aapl) / lag(aapl, k = -1) * 100
> ret_cont <- diff(log(aapl)) * 100
```

同时，我们也可以得到简单收益率的概括统计。在这里，我们使用 `coredata` 方法来表明我们仅仅关注股票价格，而非索引（日期）。

```
> summary(coredata(ret_simple))
      Min.   1st Qu.   Median     Mean 3rd Qu.     Max.
-51.86000 -1.32500  0.07901  0.12530  1.55300 13.91000
```

可以看出，最大的单日损失是-51.86%。我们还可以使用下面的命令获得这个损失发生的日期。

```
> ret_simple[which.min(ret_simple)]
2000-09-29
      -51.85888
```

上网快速搜索可以发现，这个股价的剧烈变动缘于一个盈利预警的发布。我们

可以画出直方图来加深理解日收益率的相关频率。对收益率数据进行分组时，我们可以使用 `break` 参数来指定每组的元素个数。

```
> hist(ret_simple, breaks=100, main = "Histogram of Simple Returns",
+       xlab="%")
```

我们也可以把分析限定于时间序列的一个子集 (window) 中。比如，苹果股价在 2013 年的最高点可以通过运行下面的命令的找到。

```
> aapl_2013 <- window(aapl, start = '2013-01-01', end = '2013-
12-31')
> aapl_2013[which.max(aapl_2013)]
2013-01-02
545.85
```

从风险管理的角度看，收益率分布的分位数很有意义。比如，我们使用简单的历史模拟法，可以很容易确定一天中置信水平为 99% 的在险价值 (Value-at-Risk)。

```
> quantile(ret_simple, probs = 0.01)
1%
-7.042678
```

因此，在任意给定的一天中，收益率低于 -7% 的概率只有 1%。但是如果这一天发生了这样的情形 (每年大约会发生 2.5 次)，7% 将是最小的损失量。

线性时间序列的建模与预测

线性时间序列的一类重要模型是自回归单整移动平均 (Autoregressive Integrated Moving Average, ARIMA) 模型族，它由 Box 和 Jenkins (1976 年) 提出。ARIMA 模型假定了时间序列的当前值只依赖于自身的过去值和某些误差项的过去值。

根据 Box 和 Jenkins 的研究，建立 ARIMA 模型包含了以下 3 个阶段。

1. 模型识别。

2. 模型估计。

3. 模型诊断检验。

模型识别的阶段包括了使用图方法或信息准则来确定试验模型的阶数（包含的过去值个数和过去误差项个数）。模型阶数确定之后需要估计模型参数，通常会使用最小二乘法或者极大似然方法。最后，为了检查模型可能存在的缺陷，必须仔细检查拟合的模型。这个目的可以通过保证模型残差的行为符合白噪声的特点来实现，换句话说，残差不存在线性依赖。

1.2 对英国房屋价格建模并预测

除了 zoo 包，我们还会使用到 forecast 包的一些方法。如果你还没安装它，那就需要运行下面的命令来安装 forecast 包。

```
> install.packages("forecast")
```

接着我们运行下面的命令载入类。

```
> library("forecast")
```

首先，我们在时间序列对象 zoo 中存储月度房屋价格数据（来源：全英房屋抵押贷款协会）。

```
> hp <- read.zoo("UKHP.csv", sep = ",",
+ header = TRUE, format = "%Y-%m", FUN = as.yearmon)
```

参数 FUN 对日期列调用给定的函数（as.yearmon，表示月度数据点）。为了确认指定 as.yearmon 真正存储了月度数据（每个周期 12 个子周期），我们可以查询数据序列的频率。

```
> frequency(hp)
```

```
[1] 12
```

结果表明,一个周期(称为年)中有12个子周期(称为月)。为了深入分析,我们再次计算数据的简单收益率。

```
> hp_ret <- diff(hp) / lag(hp, k = -1) * 100
```

1.2.1 模型识别和估计

我们使用 forecast 包提供的 auto.arima 函数,在一步中同时识别最优模型并估计参数。除了收益率序列(hp_ret),函数还使用了几个参数。通过指定 stationary = TRUE,我们将搜索仅仅限于平稳模型。同样地,seasonal = FALSE 将搜索限定于非平稳模型。此外,我们选择模型时,选择赤池信息量来度量模型的相对质量。

```
> mod <- auto.arima(hp_ret, stationary = TRUE, seasonal = FALSE,
+   ic="aic")
```

为了确定拟合系数的值,我们可以查询模型的输出。

```
> mod
Series: hp_ret
ARIMA(2,0,0) with non-zero mean
```

Coefficients:

	ar1	ar2	intercept
	0.2299	0.3491	0.4345
s.e.	0.0573	0.0575	0.1519

```
sigma^2 estimated as 1.105: log likelihood=-390.97
```

```
AIC=789.94   AICc=790.1   BIC=804.28
```

根据赤池信息量准则,看起来 AR(2) 模型拟合数据最好。为了视觉层次的确认,我们可以使用命令 pacf 画出偏自相关函数。图形显示了到滞后两阶的非零偏自相关,阶数为 2 的 AR 过程看来比较合适。AR(2) 模型给出了两个 AR 系

数、截距（如果模型包含 AR 项，截距实际上是均值）以及相应的标准误。在下面的例子中，因为水平为 5% 的置信区间没有包括 0，所以这些统计量都在 5% 的水平上显著。

```
> confint(mod)
                2.5 %      97.5 %
ar1          0.1174881 0.3422486
ar2          0.2364347 0.4617421
intercept 0.1368785 0.7321623
```

如果模型包含不显著的系数，我们可以使用带有 `fixed` 参数的 `arima` 函数重新估计模型，这相当于输入元素为 0 和 NA 的向量。NA 表示相应的变量系数需要估计而 0 表示相应的变量系数需要设置为 0。

1.2.2 模型诊断检查

一个快速验证模型的方法是运行下面的命令画出时间序列的诊断图。

```
> tsdiag(mod)
```

上述命令的输出在图 1-1 中显示。

标准化残差看来没有表现出波动率聚集，ACF 图中的残差自相关并不显著，还有自相关的 Ljung-Box 检验的 p 值看起来很高，综上所以残差独立的原假设不能被拒绝，因此模型看来良好。

为了评估模型对样本数据的拟合良好程度，我们可以画出原始的月回报（细的黑色实线）与拟合值（宽的红色点线）的对比图形。

```
> plot(mod$x, lty = 1, main = "UK house prices: raw data vs.
fitted + values", ylab = "Return in percent", xlab = "Date")
> lines(fitted(mod), lty = 2, lwd = 2, col = "red")
```

输出显示在图 1-2 中。

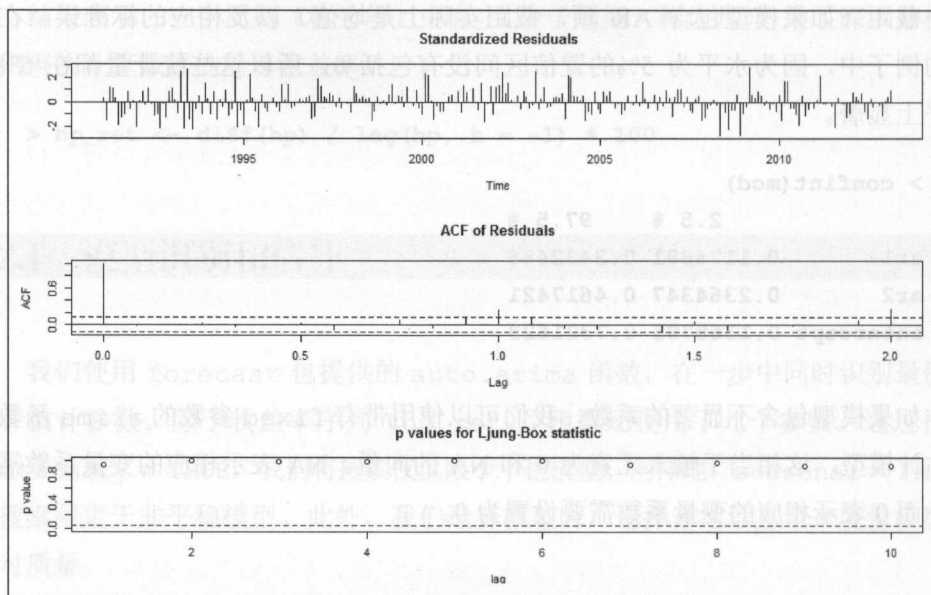


图 1-1 时间序列的诊断

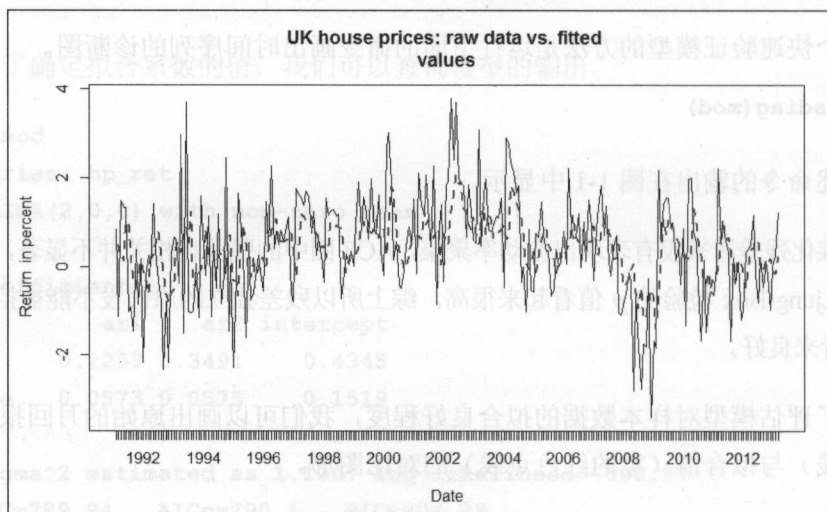


图 1-2 英国房屋数据

此外，我们可以计算精确性的常用度量。

> accuracy(mod)

ME	RMSE	MAE	MPE	MAPE	MASE
0.00120	1.0514	0.8059	-Inf	Inf	0.792980241

这个命令返回平均误差、均方误差、平均绝对误差、平均百分比误差、平均绝对值百分比误差和平均绝对比例误差。

1.2.3 预测

为了预测接下来 3 个月的月收益率 (2013 年 4~6 月)，我们使用下面的命令。

```
> predict(mod, n.ahead=3)
$pred
      Apr      May      Jun
2013 0.5490544 0.7367277 0.5439708
$se
      Apr      May      Jun
2013 1.051422 1.078842 1.158658
```

所以，我们预期在接下来的 3 个月中，平均房屋价格稍有增长，但标准误比较高，大约为 1%。为了画出带有标准误的预测，我们可以使用下面的命令。

```
> plot(forecast(mod))
```

1.3 协整

协整的思想缘于 Granger (1981 年) 提出的一个概念，后来由 Engle 和 Granger (1987 年) 加以形式化。协整的思想是指寻找非平稳时间序列之间的一个线性组合，这个线性组合是一个平稳时间序列。因此，协整方法可以用于检测非平稳时间序列 (比如价格) 之间的稳定长期关系。

航空燃油的交叉对冲

航空公司很自然需要购买航空燃油。但由于航空燃油价格的波动很剧烈，大部分航空公司会将它们对航空燃油价格变化的风险敞口对冲掉一部分。如果市场中缺乏航空燃油 OTC 产品（译注：OTC 产品指交易所场外柜台交易产品），航空公司会使用相关交易所交易的期货合约（比如，取暖油）来实现对冲。在下面的部分中，我们首先使用经典方法导出最优对冲比率，这种方法仅仅考虑两种价格之间短期波动。然后考察价格之间的长期稳定联系，进而改进最优对冲比。

首先，我们载入需要使用的包。urca 包有一些有用的方法，可以用于单位根检验和估计协整关系。

```
> library("zoo")
> install.packages("urca")
> library("urca")
```

我们导入航空燃油和取暖油的月价格。

```
> prices <- read.zoo("JetFuelHedging.csv", sep = ",",
+ FUN = as.yearmon, format = "%Y-%m", header = TRUE)
```

现在我们仅仅考虑两种商品的短期行为（月价格改变）。通过拟合一个用取暖油价格变化来解释航空燃油价格的线性模型，可以推导出两种商品的最小方差对冲比率。线性模型中的 β 系数就是最优对冲比。

```
> simple_mod <- lm(diff(prices$JetFuel) ~ diff(prices$HeatingOil)+0)
```

函数 `lm`（用于线性模型）估计了航空燃油价格变动对取暖油价格变动的最佳拟合系数。`+0` 项表示截距设置为 0，意味着航空公司不持有现金。

```
> summary(simple_mod)
```

```

Call:
lm(formula = diff(prices$JetFuel) ~ diff(prices$HeatingOil) +
    0)

Residuals:
    Min       1Q   Median       3Q      Max
-0.52503 -0.02968  0.00131  0.03237  0.39602

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
diff(prices$HeatingOil)  0.89059     0.03983   22.36  <2e-16***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0846 on 189 degrees of freedom
Multiple R-squared:  0.7257, Adjusted R-squared:  0.7242
F-statistic: 499.9 on 1 and 189 DF, p-value: < 2.2e-16

```

结果得到了方差比为 0.89059 和残差标准差为 0.0846。但是，交叉对冲并非完美无缺，推导出的对冲组合结果仍然有风险。

现在，我们通过考察航空燃油和取暖油期货价格之间存在的长期关系，尝试改进方差比。我们使用下列命令画出两个价格序列（取暖油价格用红色），通过观察图形你可能已经猜出存在着这种长期关系。

```

> plot(prices$JetFuel, main = "Jet Fuel and Heating Oil Prices",
+       xlab = "Date", ylab = "USD")
> lines(prices$HeatingOil, col = "red")

```

我们使用 Engle 和 Granger 的两步估计方法。首先，使用增强的 Dickey-Fuller 检验（augmented Dickey-Fuller test, ADF 检验）对两个序列进行单位根检验（非平稳性）。

```

> jf_adf <- ur.df(prices$JetFuel, type = "drift")
> summary(jf_adf)
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####

```

```

Test regression drift

```

```

Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
    Min       1Q   Median       3Q      Max
-1.06212 -0.05015  0.00566  0.07922  0.38086

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03050     0.02177   1.401 0.16283
z.lag.1      -0.01441     0.01271  -1.134 0.25845
z.diff.lag   0.19471     0.07250   2.686 0.00789 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.159 on 186 degrees of freedom
Multiple R-squared:  0.04099, Adjusted R-squared:  0.03067
F-statistic: 3.975 on 2 and 186 DF, p-value: 0.0204

Value of test-statistic is: -1.1335 0.9865

Critical values for test statistics:
      1pct 5pct 10pct
tau2  -3.46 -2.88 -2.57
phil   6.52  4.63  3.81

```

结果显示, 因为检验统计量值-1.1335 大于临界值-3.46, 所以在 1%的置信水平上不能拒绝非平稳 (航空燃油时间序列包含一个单位根) 的原假设。同样的结果对取暖油也成立 (检验统计量是-1.041)。

```

> ho_adf <- ur.df(prices$HeatingOil, type = "drift")
> summary(ho_adf)

```

现在我们可以继续估计静态均衡模型, 并使用 ADF 方法检验时间序列的残差是否平稳。请注意, 目前的研究序列是上一步的估计结果, 因此, 我们现在必须使用不同的临界值 [参见 Engle 和 Yoo1987 发表的论文]。

```

> mod_static <- summary(lm(prices$JetFuel ~ prices$HeatingOil))
> error <- residuals(mod_static)
> error_cadf <- ur.df(error, type = "none")
> summary(error_cadf)

```

得到的检验统计量是-8.912, 而规模为 200 的样本在 1%的置信水平上的临界值

为-4.00。所以，我们拒绝非平稳的原假设。因此我们发现了两个协整变量并且可以进行第二步，两个协整变量意味着一个误差修正模型（ECM）。ECM 是一个动态模型，刻画了系统如何（以及多快）返回之前估计的静态均衡，这个静态均衡存储在变量 `mod_static` 中。

```
> djf <- diff(prices$JetFuel)
> dho <- diff(prices$HeatingOil)
> error_lag <- lag(error, k = -1)
> mod_ecm <- lm(djf ~ dho + error_lag)
> summary(mod_ecm)
```

Call:

```
lm(formula = djf ~ dho + error_lag + 0)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.19158	-0.03246	0.00047	0.02288	0.45117

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
Dho	0.90020	0.03238	27.798	<2e-16 ***
error_lag	-0.65540	0.06614	-9.909	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06875 on 188 degrees of freedom

Multiple R-squared: 0.8198, Adjusted R-squared: 0.8179

F-statistic: 427.6 on 2 and 188 DF, p-value: < 2.2e-16

通过考察航空燃油价格和取暖油价格之间存在的长期联系（协整），对冲比现在稍微提高了一点（0.90020），并且残差标准差显著地降低了（-0.65540）。误差项的系数为负（-0.65540）：两个价格之间原先较大的偏差有所修正，价格向它们的长期稳定关系移动。

1.4 波动率建模

正如我们之前所见，ARIMA 模型常常用于过程的过去值已知时的条件期望建

模。过去值已知的过程的条件方差是常数。真实世界的金融时间序列存在着波动性聚集和其他特点,换句话说,突发波动率打破了相对稳定的时期。

在这一节中,我们来考查 GARCH 时间序列模型。GARCH 模型研究真实世界的(金融)时间序列的这个典型化事实——波动性聚集,并进一步运用这些模型预测在险价值(Value at Risk, VaR)。

1.4.1 风险管理的波动率预测

金融机构使用 VaR 来度量他们的活动风险,通常在 10 个工作日范围内计算置信水平为 99% 的临界值。这意味着在这 10 天内,只有 1% 的时间会出现预期损失超过临界值。

我们载入 zoo 包并导入英特尔公司的月收益率数据,时间范围从 1973 年 1 月~2008 年 12 月。

```
> library("zoo")
> intc <- read.zoo("intc.csv", header = TRUE,
+   sep = ",", format = "%Y-%m", FUN = as.yearmon)
```

1.4.2 检验 ARCH 效应

收益率图形表明,在月收益率数据中可能存在 ARCH 效应。

```
> plot(intc, main = "Monthly returns of Intel Corporation",
+   xlab = "Date", ylab = "Return in percent")
```

上面命令的输出在图 1-3 中显示。

我们可以使用统计假设检验来验证自己的想法。两种常用检验如下。

- 用于平方收益率(波动率的一种代理)自相关的 Ljung-Box 检验。

- Engle (1982 年) 提出的拉格朗日乘子 (LM) 检验。

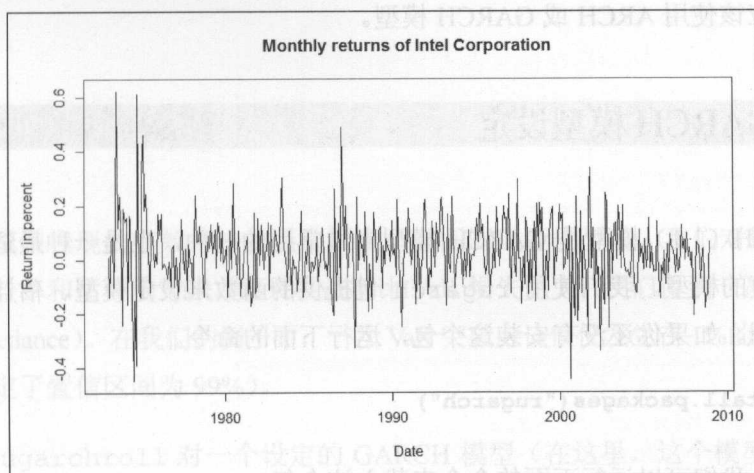


图 1-3 英特尔公司的月收益

首先, 我们运行下面的命令, 从而在平方收益率的前 12 阶滞后值上执行 Ljung-Box 检验。

```
> Box.test(coredata(intc^2), type = "Ljung-Box", lag = 12)
```

Box-Ljung test

```
data: coredata(intc^2)
```

```
X-squared = 79.3451, df = 12, p-value = 5.502e-12
```

我们可以在 1% 的置信水平上拒绝原假设, 原假设是平方收益率中不存在自相关。或者, 我们可以使用 FinTS 包的 LM 检验, 它输出相同的结果。

```
> install.packages("FinTS")
```

```
> library("FinTS")
```

```
> ArchTest(coredata(intc))
```

```
ARCH LM-test; Null hypothesis: no ARCH effects
```

```
data: coredata(intc)
```

```
Chi-squared = 59.3647, df = 12, p-value = 2.946e-08
```

两种检验都确定了英特尔的月收益率中存在 ARCH 效应。因此,收益率时间序列的建模应该使用 ARCH 或 GARCH 模型。

1.4.3 GARCH 模型设定

GARCH (1,1) 模型是 GARCH 模型中最常用的一种,也是一种最适于金融时间序列建模的模型。我们使用 rugarch 包提供的函数来设定模型、估计参数、回测以及预测。如果你还没有安装这个包,运行下面的命令。

```
> install.packages("rugarch")
```

然后,我们可以运行下面的命令来载入这个包。

```
> library("rugarch")
```

首先,我们需要使用函数 ugarchspec 设定模型。对于一个 GARCH (1,1) 模型,我们需要设置 garchOrder 为 c(1,1)。而且均值模型 (mean.model) 是一个白噪声过程,因此等同于 armaOrder = c(0,0)。

```
> intc_garch11_spec <- ugarchspec(variance.model = list(
+   garchOrder = c(1, 1)),
+   mean.model = list(armaOrder = c(0, 0)))
```

1.4.4 GARCH 模型估计

通过 ugarchfit 函数设定模型,输入为收益率数据,就可以用极大似然方法精确拟合模型的系数。

```
> intc_garch11_fit <- ugarchfit(spec = intc_garch11_spec,
+   data = intc)
```

其他参数的使用请参见 ugarchfit 的帮助文档。拟合模型的输出 (使用命令

intc_garch11_fit) 展示了有用信息, 比如最优参数值、对数似然函数值, 以及信息准则。

1.4.5 回测风险模型

检验模型表现的有效方法是历史回测。在回测风险模型时, 我们对比整个时期的真实收益率和估计的 VaR。如果收益率比 VaR 损失更大, 我们得到一次 VaR 突破 (VaR exceedance)。在我们的例子中, 一次 VaR 突破应该仅仅发生在 1% 的情形中 (因为我们设定了置信区间为 99%)。

函数 ugarchroll 对一个设定的 GARCH 模型 (在这里, 这个模型是 intc_garch11_spec) 执行历史回测。我们指定回测如下。

- 使用的收益率数据存储在 zoo 对象 intc 中。
- 回测的起始期 (n.start) 应该是序列开始 (就是 1983 年 1 月) 时的前 120 个月。
- 每月都需要重新估计模型 (refit.every = 1)。
- 我们使用移动 (moving) 窗口来估计。
- 我们使用一个混合 (hybrid) 的解决方法。
- 我们希望计算 VAR 在 99% 的尾部置信水平上 (VaR.alpha = 0.01) 的临界值 (calculate.VaR = TRUE)。
- 我们希望保留估计的系数 (keep.coef = TRUE)。

下面的命令显示了满足上述所有要求的回测。

```
> intc_garch11_roll <- ugarchroll(intc_garch11_spec, intc,
+   n.start = 120, refit.every = 1, refit.window = "moving",
+   solver = "hybrid", calculate.VaR = TRUE, VaR.alpha = 0.01,
+   keep.coef = TRUE)
```


我们可以使用 `report` 函数检查回测报告。通过把这个参数的 `type` 参数设定为 `VaR`，这个函数对突破值执行无条件和有条件覆盖检验。`VaR.alpha` 是尾部概率，`conf.level` 是置信区间，条件覆盖的假设检验基于此建立。

```
> report(intc_garch11_roll, type = "VaR", VaR.alpha = 0.01,
+conf.level = 0.99)
```

VaR Backtest Report

```
Model: sGARCH-norm
```

```
Backtest Length: 312
```

```
Data:
```

```
alpha: 1%
```

```
Expected Exceed: 3.1
```

```
Actual VaR Exceed: 5
```

```
Actual %: 1.6%
```

Unconditional Coverage (Kupiec)

```
Null-Hypothesis: Correct Exceedances
```

```
LR.uc Statistic: 0.968
```

```
LR.uc Critical: 6.635
```

```
LR.uc p-value: 0.325
```

```
Reject Null: NO
```

Conditional Coverage (Christoffersen)

```
Null-Hypothesis: Correct Exceedances and
```

```
Independence of Failures
```

```
LR.cc Statistic: 1.131
```

```
LR.cc Critical: 9.21
```

```
LR.cc p-value: 0.568
```

```
Reject Null: 0
```

Kupiec 的无条件覆盖方法比较了给定 VaR 尾部概率时，预期突破值数目和实际突破值数目，而 Christoffersen 检验方法则是一种对无条件覆盖和突破值的独立性的联合检验。在我们的例子中，尽管预期突破有 3 次但实际发生了 5 次，我们不能拒绝突破是正确并且独立的原假设。

回测表现的图形也很容易生成。首先，使用 `ugarchroll` 对象的精确预测 VaR 创建一个 `zoo` 对象。

```
> intc_VaR <- zoo(intc_garch11_roll@forecast$VaR[, 1])
```

我们仍然使用这个“zoo”对象，通过 `rownames`（年和月）重写这个对象的 `index` 属性。

```
> index(intc_VaR) <- as.yearmon(rownames(intc_garch11_roll@forecast$VaR))
```

对同时存储在 `ugarchroll` 对象中的真实收益率，我们加以同样的处理。

```
> intc_actual <- zoo(intc_garch11_roll@forecast$VaR[, 2])
```

```
> index(intc_actual) <-
```

```
as.yearmon(rownames(intc_garch11_roll@forecast$VaR))
```

现在，我们可以使用下面的命令，画出 VaR 对比英特尔真实收益率的图形。

```
> plot(intc_actual, type = "b", main = "99% 1 Month VaR Backtesting",
+       xlab = "Date", ylab = "Return/VaR in percent")
> lines(intc_VaR, col = "red")
> legend("topright", inset=.05, c("Intel return", "VaR"), col =
+       c("black", "red"), lty = c(1,1))
```

图 1-4 中显示了上述命令行的输出。

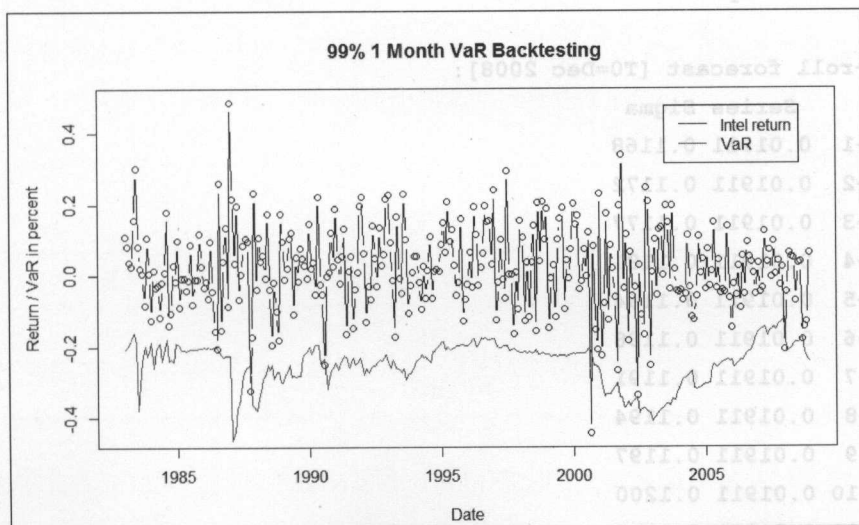


图 1-4 99% 1 月 VaR 回测

1.4.6 预测

我们现在有理由相信风险模型运行正常，我们也可以生成 VaR 预测。函数 `ugarchforecast` 选取以下两个参数，一个是拟合的 GARCH 函数 (`intc_garch11_fit`)，另一个是应该产生预测的周期数 (`n.ahead = 12`，即 12 个月)。

```
> intc_garch11_fcst <- ugarchforecast(intc_garch11_fit, n.ahead = 12)
```

我们可以通过查询以下命令行显示的预测对象，来预期未来结果。

```
> intc_garch11_fcst
*-----*
*          GARCH Model Forecast          *
*-----*
Model: sGARCH
Horizon: 12
Roll Steps: 0
Out of Sample: 0

Conditional Coverage (Christoffersen)
0-roll forecast [T0=Dec 2008]:
```

	Series	Sigma
T+1	0.01911	0.1168
T+2	0.01911	0.1172
T+3	0.01911	0.1177
T+4	0.01911	0.1181
T+5	0.01911	0.1184
T+6	0.01911	0.1188
T+7	0.01911	0.1191
T+8	0.01911	0.1194
T+9	0.01911	0.1197
T+10	0.01911	0.1200
T+11	0.01911	0.1202
T+12	0.01911	0.1204

波动率 (sigma) 的一步预期是 0.1168。因为我们假定了正态分布, 置信水平为 99% 的 VaR 可以使用标准正态分布的 99% 分位数 (输入 `qnorm(0.99)`) 来计算。因此对下一个周期, 一个月的 99% VaR 就是 $qnorm(0.99) * 0.1168 = 0.2717$ 。结果, 月收益率高于 -27% 的概率是 99%。

1.5 小结

在本章中, 我们运用 R 分析了时间序列选择的一些问题。本章涵盖了时间序列数据的不同表达方法, 使用 ARMA 模型预测房屋价格, 使用协整关系改进基本的最小方差对冲比, 以及使用 GARCH 模型进行风险管理。在下一章, 你将学习如何使用 R 来构建最优投资组合。

图 1-5 展示了不同置信水平下的 VaR 值。图中显示了三个置信水平: 99% (虚线), 95% (点线), 和 90% (实线)。随着置信水平的提高, VaR 值也随之增加。

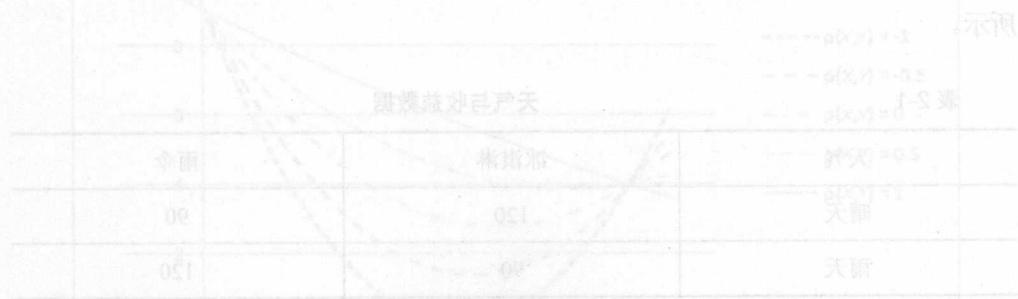


图 1-5 展示了不同置信水平下的 VaR 值。图中显示了三个置信水平: 99% (虚线), 95% (点线), 和 90% (实线)。随着置信水平的提高, VaR 值也随之增加。

$$[20.0 - 1 - 0.01(0.02 \times 2.0 + 0.01 \times 2.0)] \times$$

图 1-5 展示了不同置信水平下的 VaR 值

第 2 章

投资组合优化

到现在为止，我们已经熟悉了 R 语言的基础。我们知道如何去分析数据、调用它的内置函数并把它们运用到我们选择的时间序列分析问题上。在本章中，我们既运用这些知识，又通过一种重要的实践应用来扩展这种知识，即投资组合分析，换句话说也叫证券选择。这一节涵盖了投资组合优化背后的思想：数学模型和理论求解。为了提高编程技巧，我们使用真实数据解决一个现实中的问题，并逐行实施算法。同时，我们也在相同的数据集上使用预先写好的 R 包。

想象我们生活在一个热带岛屿，只有 100 美元可以投资。这个岛上的投资机会相当有限。我们可以把全部资金投资到冰淇淋上或雨伞上。收益取决于天气，如表 2-1 所示。

表 2-1 天气与收益数据

天气	冰淇淋	雨伞
晴天	120	90
雨天	90	120

假定天气是晴天或是雨天的概率相同。如果我们不能预知天气或者改变天气，这两种选择的概率显然相等，我们投资于其中任何一种，都会得到 5% 的预期收益率 $[(0.5 \times 120 + 0.5 \times 90) / 100 - 1 = 0.05]$ 。

如果我们可以把资金分配在冰淇淋和雨伞之间，那该如何划分资金？我们应该在两种备选各投资 50 美元。无论会发生什么，我们会在一种资产上赚到 45 美元而在另一种资产上赚到 60 美元，因此，这个组合没有风险。预期收益率仍然是 5%，但因为 $(45+60)/100-1=0.05$ ，现在的收益获得了保障。

这个例子抓住了投资组合优化的主要概念（正是因为这个理论，哈里·马科维茨在 1990 年获得诺贝尔奖）。通过考虑投资产品之间的相关性，我们可以在保持想要的预期收益率不变的前提下减少投资组合的风险（在这个例子里由方差来表示）。

为了得到精确的数学表达式，令 X 和 Y 是随机变量，各自方差有限并分别为 σ_x^2 和 σ_y^2 。它们的凸组合或仿射组合的方差显示在下面的二次函数中。

$$f(\alpha) = \text{Var}(\alpha X + (1-\alpha)Y) = \alpha^2 \sigma_x^2 + (1-\alpha)^2 \sigma_y^2 + 2\alpha(1-\alpha)\text{Cov}(X, Y)$$

对于它们相关系数的不同值，这个二次函数如图 2-1 所示。

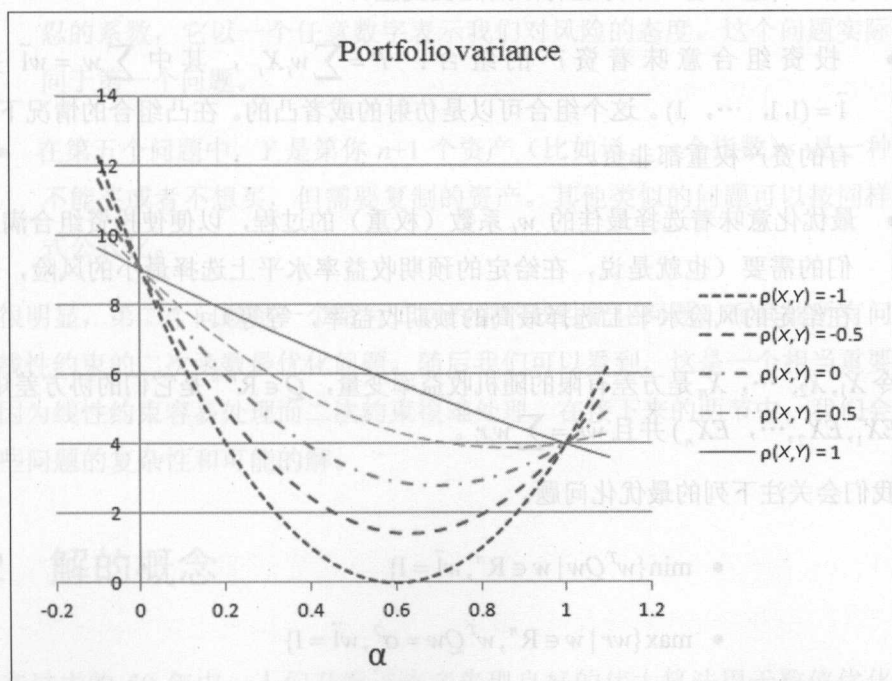


图 2-1 投资组合的方差

当且仅当 X 和 Y 的相关系数为 -1 ，并且 X 和 Y 的方差相同时，方差（作为风险的一种度量）可以完全消除掉（译者注，原文此处有误）。否则，我们会在后面的“定理（拉格朗日）”这一节中看到，权重最优的投资组合的方差取决于（以一种完全不平凡的方式）所有 3 个参数，分别是 σ_x^2 、 σ_y^2 以及 $Cov(X, Y)$ 。

2.1 均方差模型

马科维茨[Markowitz, H.M. (March 1952)]提出的均方差模型，其实是冰淇淋/雨伞交易模型在高维上的体现。为了得到数学公式，我们需要一些相关的定义。

定义的解释如下。

- 资产 X_i 意味着一个方差有限的随机变量。
- 投资组合意味着资产的组合： $P = \sum w_i X_i$ ，其中 $\sum w_i = w\bar{1}$ 并且 $\bar{1} = (1, 1, \dots, 1)$ 。这个组合可以是仿射的或者凸的。在凸组合的情况下，所有的资产权重都非负。
- 最优化意味着选择最佳的 w_i 系数（权重）的过程，以便使投资组合满足我们的需要（也就是说，在给定的预期收益率水平上选择最小的风险，或者在给定的风险水平上选择最高的预期收益率，等等）。

令 X_1, X_2, \dots, X_n 是方差有限的随机收益率变量， $Q \in \mathbb{R}^{n \times n}$ 是它们的协方差矩阵， $r = (EX_1, EX_2, \dots, EX_n)$ 并且 $wr = \sum w_i r_i$ 。

我们会关注下列的最优化问题。

$$\bullet \min \{w^T Q w \mid w \in \mathbb{R}^n, w\bar{1} = 1\} \quad (1)$$

$$\bullet \max \{wr \mid w \in \mathbb{R}^n, w^T Q w = \sigma^2, w\bar{1} = 1\} \quad (2)$$

$$\bullet \min \{w^T Q w \mid w \in \mathbb{R}^n, w\bar{1} = 1, wr = \mu\} \quad (3)$$

$$\bullet \max\{wr - \lambda w^T Qw \mid w \in \mathbb{R}^n, w\bar{1} = 1\} \quad (4)$$

$$\bullet \min\{\sigma^2(w^T X - Y) \mid w \in \mathbb{R}^n, w\bar{1} = 1\} \quad (5)$$

很明显, $w^T Qw$ 是投资组合的方差, wr 是预期收益率。对于权重和, 我们有 $w\bar{1} = 1$, 这意味着我们愿意投资一个单位的资金。(我们也可以考虑增加 $w \geq 0$ 的条件, 这意味着不允许做空。) 下面各点详细地解释了问题。

- 第一个问题是寻找风险最小的投资组合。如果不考虑无风险资产, 答案可以是非平凡的。
- 第二个问题是在给定的方差水平上选择最大的预期收益率。
- 一个稍微不同的方法是在一个想要的预期收益率水平上寻找方差最小的投资组合。
- 第四个问题是最大化一个简单的效用函数, $\text{return} - \lambda \times \text{risk}$, 其中 λ 是风险容忍的系数, 它以一个任意数字表示我们对风险的态度。这个问题实际上等同于第一个问题。
- 在第五个问题中, Y 是第你 $n+1$ 个资产 (比如说, 一个指数), 是一种我们不能买或者不想买, 但需要复制的资产。其他类似的问题可以按同样的方式公式化。

很明显, 第二个问题是一个带有二次约束的线性优化问题, 而其他所有问题是带有线性约束的二次函数最优化问题。随后我们可以看到, 这是一个相当重要的区别, 因为线性约束容易处理而二次约束很难处理。在接下来的两节中, 我们会专注于这些问题的复杂性和可能的解。

2.2 解的概念

在过去的 50 年中, 人们开发了许多表现良好的伟大算法用于数值优化, 尤其用于二次函数的情形。正如我们在上一节中看到的, 我们只有二次函数和约束。

所以这些方法（它们也在 \mathbf{R} 中实现）可以用于最糟的情形（如果没有任何更好的情形）。

然而，数值优化的细节讨论超出了本书范围。幸运的是，在线性约束和二次函数约束的特殊情况下，这些方法不是并不必要。我们可以使用 18 世纪提出的拉格朗日定理。

2.1 均方差模型

（拉格朗日）定理

如果 $f: \mathbf{R}^n \rightarrow \mathbf{R}$ 和 $g=(g_1, g_2, \dots, g_m): \mathbf{R}^n \rightarrow \mathbf{R}^m$ （其中 $m < n$ ）有连续的偏导数，并且 $a \in \{g(x)=0\}$ 是使得 $g(x)=0$ 的 $f(x)$ 的相对极值点，其中限定 $\text{rank}(g'(a))=m$ 。

那么，这里存在系数 $\lambda_1, \lambda_2, \dots, \lambda_m$ 使得 $f'(a) + \sum \lambda_i g'_i(a) = 0$ 。换句话说，这个函数 $L = f - \sum \lambda_i g_i: \mathbf{R}^{m+n} \rightarrow \mathbf{R}$ 所有的偏导数都是 0（Bertsekas Dimitri P. (1999)）。

在我们的例子里，这个条件也是充分的。二次函数的偏导数是线性的，所以最优化产生了求解一个线性系统方程的问题，（不像数值优化）这是一个高中水平的任务。

我们来看一下，如何运用它来解决第三个问题。

$$\min \{w^T Q w \mid w \in \mathbf{R}^n, w\bar{1} = 1, w\bar{r} = \mu\}$$

可以表明，这个问题等价于下面的线性方程系统。

$$\begin{bmatrix} Q & \bar{1} & \bar{r} \\ \bar{1} & 0 & 0 \\ \bar{r} & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \mu \end{bmatrix}$$

（两行和两列加入到协方差矩阵中，所以我们有条件同时决定两个拉格朗日乘子。）我们可以期望这个系统有唯一解。

需要强调的是，我们利用拉格朗日定理得到的结果已经不再是一个最优化问题。就像是在一维中，最小化二次函数需要求导数，但从复杂性的角度看，

一个线性系统的方程是平凡的。现在，我们来看看如何处理收益率最大化的问题。

$$\max\{w^T r \mid w \in \mathbb{R}^n, w^T Q w = \sigma, w^T \mathbf{1} = 1\}$$

容易看出，拉格朗日函数对于 λ 的导数是约束本身。

为了看出这一点，我们取 L 的导数。

$$L := f + \sum \lambda_i g_i$$

$$\partial L / \partial \lambda_i = g_i$$

由此产生了一个非线性方程，它的求解与其说是科学不如说是艺术。

2.3 使用真实数据

投资组合的最优化完全整合在随后我们会讨论的多个 R 包中，知道这一点非常有用。但是，跑步之前最好先会走路。因此，我们从一个简单的自制 R 函数开始，我们把它一行行地在下面列出来。

```
minvariance <- function(assets, mu = 0.005) {
  return <- log(tail(assets, -1) / head(assets, -1))
  Q <- rbind(cov(return), rep(1, ncol(assets)),
             colMeans(return))
  Q <- cbind(Q, rbind(t(tail(Q, 2)), matrix(0, 2, 2)))
  b <- c(rep(0, ncol(assets)), 1, mu)
  solve(Q, b)
}
```

这就是我们在“(拉格朗日)定理”这一节讨论的算法的直接运用。

为了做出示范，我们从 **Quandl** 超集中选取了一些 IT 股票的价格(http://www.quandl.com/USER_1KR/1KT)，**Quandl** 是一种公共服务，提供了轻松获取大批定量数据的途径。这个超链接 [http://www.quandl.com/api/v1/datasets/USER_1KR/

1KT.csv)] 指向一个可下载的逗号分隔值 (CSV) 文件, 可以保存在硬盘上, 并通过 read.csv 导入到 R 中, 此外还有一种更直观的方法来完成这种操作, 就是借助于包括在上述超链接中的键的帮助。

```
> library(Quandl)
> IT <- Quandl('USER_1KR/1KT',
+             start_date = '2008-01-01', end_date = '2012-12-31')
Warning message:
In Quandl("USER_1KR/1KT", start_date = "2008-01-01", end_date =
"2012-12-31"):
```

如果你没有使用认证令牌, 可能会出现上面的警告信息。请访问 <http://www.quandl.com/help/r>, 或者每天仅从 Quandl 下载 10 个数据集。

```
> str(IT)
'data.frame': 1259 obs. of 6 variables:
 $ Date: Date, format: "2008-01-02" "2008-01-03" ...
 $ AAPL: num 199 195 191 181 180 ...
 $ GOOG: num 693 685 680 654 653 ...
 $ MSFT: num 35.8 35.2 35.2 34.5 34.7 ...
 $ IBM : num 109 105 104 100 100 ...
 $ T   : num 41.5 41.2 41 41.1 41.3 ...
```

因此, 我们载入 Quandl 包, 它提供的 Quandl 函数需要以下这几个参数。

- 第一个参数 (code=USER_1KR/1KT) 是数据集在 Quandl 上的代码。
- start_date 和 end_date 参数可以任意指定我们感兴趣的时间段, 在这里我们设置为从五年前到现在。
- 更多的选择请参考?Quandl。例如, 可以使用 type 导入已经存储在某些时间序列对象中的数据, 来代替最初的 data.frame。

在新创建的 IT 变量上运行 str 命令, 结果显示了 R 对象的内部结构, 它当前包括一个 Date 域和 5 个数值格式的资产价格。

将价格从 IT 中指派给 assets 之后（不包括第一列 Date），我们来逐行运行之前的 minvariance 函数的主体部分。首先，我们计算资产收益率，除了第一个值，每个值都除以自身的前一个值，再对每个商计算 log。

```
> assets <- IT[, -1]
> return <- log(tail(assets, -1) / head(assets, -1))
```

请注意，收益率也可以直接由 timeSeries 包的 returns 函数计算，为了教学的目标，我们在这里不调用它。为了确认命令做了什么，我们来检查新创建的变量的前一些值。

```
> head(return)
      AAPL      GOOG      MSFT      IBM      T
2 -0.019560774 -0.011044063 -0.0160544217 -0.038916144 -0.0072534167
3 -0.020473237 -0.008161516 -0.0008521517 -0.008429976 -0.0043774389
4 -0.054749384 -0.038621208 -0.0183544011 -0.036242948 0.0007309051
5 -0.006142967 -0.001438475 0.0046202797 -0.001997005 0.0051014322
6 -0.050317921 -0.035793820 -0.0396702524 -0.023154566 -0.0514590970
7 0.036004806 0.023482511 0.0292444412 -0.003791959 -0.0123204844
```

接着，我们来建立拉格朗日定理指定的线性等式系统的左侧：

$$\begin{bmatrix} Q \\ \bar{1} \\ \bar{r} \end{bmatrix}$$

，其中我们

按行 (rbind) 把协方差矩阵 (cov)，根据数据集列数 (ncol) 重复 (rep) 次数的 1 的组合向量，以及收益率的均值 (colMeans) 结合在一起。

```
> Q <- rbind(cov(return), rep(1, ncol(assets)), colMeans(return))
```

最后，结果如下。

```
> round(Q, 5)
      AAPL      GOOG      MSFT      IBM      T
AAPL 0.00063 0.00034 0.00025 0.00023 0.00022
GOOG 0.00034 0.00046 0.00023 0.00019 0.00018
MSFT 0.00025 0.00023 0.00034 0.00018 0.00018
```

```

IBM 0.00023 0.00019 0.00018 0.00024 0.00016
T 0.00022 0.00018 0.00018 0.00016 0.00028
1.00000 1.00000 1.00000 1.00000 1.00000
0.00075 0.00001 -0.00024 0.00044 -0.00018

```

请注意，为了结果易读，我们已经将运算结果保留至5位小数。还有请注意，微软（Microsoft，MSFT）和美国电话电报公司（AT&T）的均值都为负。现在，我们也需要将矩阵 `tail` 的最后两行结合在一起生成新列放在矩阵左部，为了拉格朗日定理（ 2×2 矩阵）指定的线性系统的完整性，再将其他部分设置为零。

```

> Q <- cbind(Q, rbind(t(tail(Q, 2)), matrix(0, 2, 2)))
> round(Q, 5)
      AAPL      GOOG      MSFT      IBM      T
AAPL 0.00063 0.00034 0.00025 0.00023 0.00022 1 0.00075
GOOG 0.00034 0.00046 0.00023 0.00019 0.00018 1 0.00001
MSFT 0.00025 0.00023 0.00034 0.00018 0.00018 1 -0.00024
IBM 0.00023 0.00019 0.00018 0.00024 0.00016 1 0.00044
T 0.00022 0.00018 0.00018 0.00016 0.00028 1 -0.00018
1.00000 1.00000 1.00000 1.00000 1.00000 0 0.00000
0.00075 0.00001 -0.00024 0.00044 -0.00018 0 0.00000

```

在默认情形下，`mu` 是 0.005（在最小方差函数的参数中指定）。这是线性系统

右侧向量 $\begin{bmatrix} 0 \\ 1 \\ \mu \end{bmatrix}$ 的最后一个值。

```

> mu <- 0.005
> b <- c(rep(0, ncol(assets)), 1, mu)
> b
[1] 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.005

```

在成功建立线性系统的各个部分之后，你只剩下了求解的任务。

```

> solve(Q, b)
      AAPL      GOOG      MSFT      IBM      T
2.3130600636 -1.0928257246 -2.7830264740 4.9871895547 -2.4243974197

```

```
-0.0001254637 -1.2082468413
```

上面的代码等价于一步运行这个函数，它会取数据集和最优的可选收益率作为参数。为了在想要的预期收益率水平下获得最小方差，计算结果给出了最优权重和朗格朗日乘子。

```
> minvariance(IT[, -1])
      AAPL      GOOG      MSFT      IBM      T
2.3130600636 -1.0928257246 -2.7830264740 4.9871895547 -2.4243974197
-0.0001254637 -1.2082468413
```

注意，在 Microsoft 和 AT&T 的股票前面，Google 的最优化头寸为做空。我们可以使用这个结果进而寻找最优化问题的完整解，借助其他软件和“write.csv”函数，这个求解过程还可以进一步深化。除了在给定的收益率水平上计算最小方差，我们还可以对更大范围的收益率求解最小方差。如果我们修正这份代码，可以得到如下的结果。

```
frontier <- function(assets) {
  return <- log(tail(assets, -1) / head(assets, -1))
  Q <- cov(return)
  n <- ncol(assets)
  r <- colMeans(return)
  Q1 <- rbind(Q, rep(1, n), r)
  Q1 <- cbind(Q1, rbind(t(tail(Q1, 2)), matrix(0, 2, 2)))
  rbase <- seq(min(r), max(r), length = 100)
  s <- sapply(rbase, function(x) {
    y <- head(solve(Q1, c(rep(0, n), 1, x)), n)
    y %*% Q %*% y
  })
  plot(s, rbase, xlab = 'Return', ylab = 'Variance')
}
```

除了在资产的最小收益率和最大收益率之间 (seq) 取过一个不同的收益率值 (length = 100)，和除了计算了最优投资组合的方差，这份代码与之前的版本一样。因此，我们可以画出收益率-方差对 (s 和 rbase) 来表示问题的解。结果显示在图 2-2 中。

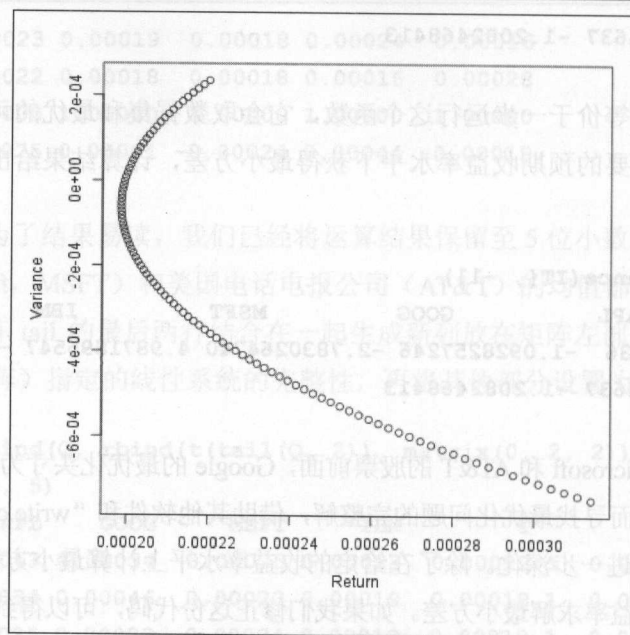


图 2-2 最优化问题的解

在方差-收益率平面上，理想的收益率-最小方差曲线叫作投资组合前沿（**Portfolio Frontier**）。忽略它向下方倾斜的部分（同样的方差可以用更高的预期收益率达到），我们得到了有效前沿（Efficient Frontier），毫无疑问必须选择有效前沿上的组合。

众所周知，两个给定的收益率水平就足以计算投资组合前沿，把得到的投资组合连接起来就得到了整个前沿。

相似的结果可以通过 R 包中的一些内置函数来计算，无需太多代码。例如，**fPortfolio** 包提供了一组有用方法，适合于 **timeSeries** 对象的应用问题。为了完成这个目的，我们必须把最初的 IT 数据集资产列转换为一个根据第一列定义的 **timeSeries** 对象。

```
> library(timeSeries)
> IT <- timeSeries(IT[, 2:6], IT[, 1])
```

就像我们在均值-方差函数中所做的那样，通过对每一个元素除以前一个值再计算对数，收益率可以定义为时间序列，但是，一些有用的时间序列命令（比如 **lag**）可以更轻松地实现这个目的。


```
> log(lag(IT) / IT)
```

或者，使用其他内置函数可以是更简单的方式。

```
> IT_return <- returns(IT)
```

现在，我们已经有了一个时间序列对象，因此绘出收益率很容易。

```
> chart.CumReturns(IT_return, legend.loc = 'topleft', main = '')
```

IT_return 中 5 个股票的收益率看上去和图 2-3 所示的一样。

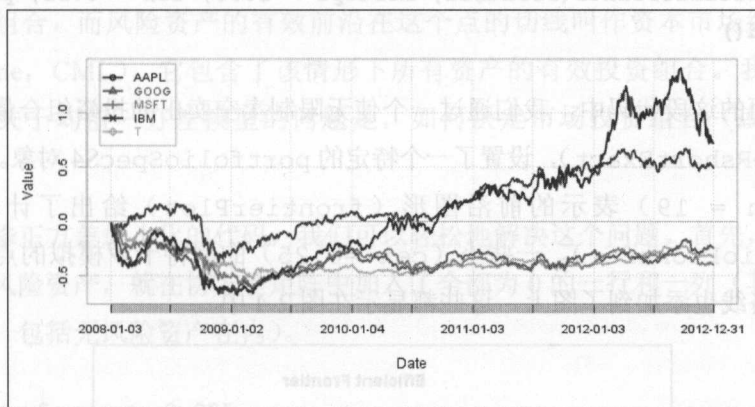


图 2-3 收益率序列

通过绘出 portfolioFrontier 的结果，可以通过交互方式画出上述的前沿图像（图 2-3）。

```
> library(fPortfolio)
```

```
> plot(portfolioFrontier(IT_return))
```

Make a plot selection (or 0 to exit):

- 1: Plot Efficient Frontier
- 2: Add Minimum Risk Portfolio
- 3: Add Tangency Portfolio
- 4: Add Risk/Return of Single Assets
- 5: Add Equal Weights Portfolio

```

6: Add Two Asset Frontiers [LongOnly Only]
7: Add Monte Carlo Portfolios
8: Add Sharpe Ratio [Markowitz PF Only]

```

为了模仿我们在上述代码中实现的内容，我们绘出带有卖空约束的前沿图像。

```

> Spec = portfolioSpec()
> setSolver(Spec) = "solveRshortExact"
> Frontier <- portfolioFrontier(as.timeSeries(IT_return),
+                               Spec, > constraints = "Short")
> frontierPlot(Frontier, col = rep('orange', 2), pch = 19)
> monteCarloPoints(Frontier, mcSteps = 1000, cex = 0.25, pch = 19)
> grid()

```

在上面的这段代码中，我们通过一个使无限制卖空卖出的投资组合最优化的函数 (solveRshortExact)，设置了一个特定的 portfolioSpecS4 对象。通过橙色圆点 (pch = 19) 表示的前沿图形 (frontierPlot) 给出了计算的结果 (portfolioFrontier)。一些小 (cex = 0.25) 的蒙特卡罗模拟的点以及作为背景的网格线也添加到了图上，这些都显示在图 2-4 中。

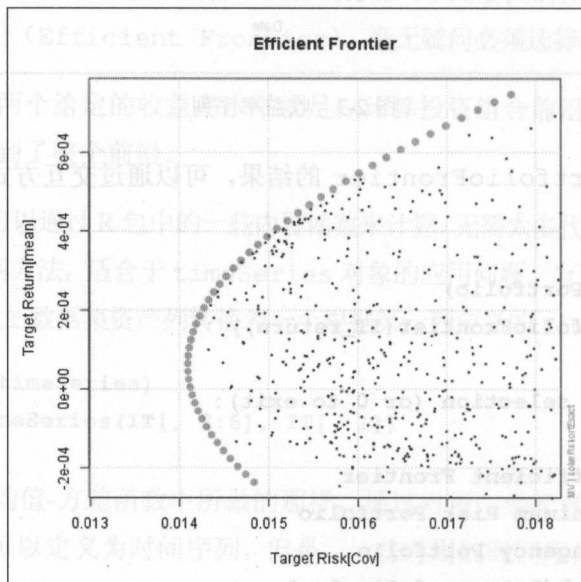


图 2-4 有效前沿

2.4 切线组合和资本市场线

当组合中加入一个无风险资产 R ，会发生什么？如果 $\sigma_R=0$ 并且 X 是任意的一个风险投资组合，那么 $\text{Var}(\alpha R + (1-\alpha)X) = (1-\alpha)^2 \text{Var}(X)$ ，并且显然也有 $E(\alpha R + (1-\alpha)X) = \alpha E(R) + (1-\alpha)E(X)$ 。这意味着这些组合在均值-标准差平面上形成了一条直线。位于这条直线上的任何投资组合都可以通过投资于 R 和 X 来得到。很明显， X 的最佳选择位于这条直线与有效边界的切点。这个切点叫作市场组合或者切点组合，而风险资产的有效前沿在这个点的切线叫作资本市场线（Capital Market Line, CML），它包含了该情形下所有资产的有效投资组合。我们解决的最后一个关于均值-方差模型的问题是，如何决定市场投资组合（或等价的，CML）。

通过修正方差最小化的代码，我们可以轻松地解决这个问题。首先，如果我们想加入无风险资产，就在协方差矩阵中加入了全部为 0 的一行和一列（其中 n 是总资产个数，包括无风险资产在内）。

```
> n <- 6; mu <- 0.005
> Q <- cbind(cov(return), rep(0, n - 1))
> Q <- rbind(Q, rep(0, n))
```

并在收益率向量中加入无风险资产。

```
> r <- c(colMeans(return), rf)
```

然后，我们可以使用新的协方差矩阵和新的收益率向量来决定最优投资组合的权重，接着基于“使用真实数据工作”这一节描述的 minvariance 代码来移除第 n 个资产。

```
> Q <- rbind(Q, rep(1, n), r)
> Q <- cbind(Q, rbind(t(tail(Q, 2)), matrix(0, 2, 2)))
```

```
> b <- c(rep(0, n), 1, mu)
```

得到下面的中间结果。

```
> round(Q, 6)
```

	AAPL	GOOG	MSFT	IBM	T	r
AAPL	0.000630	0.000338	0.000249	0.000233	0.000218	0.000748
GOOG	0.000338	0.000462	0.000226	0.000186	0.000182	0.000008
MSFT	0.000249	0.000226	0.000341	0.000178	0.000177	-0.000236
IBM	0.000233	0.000186	0.000178	0.000240	0.000157	0.000439
T	0.000218	0.000182	0.000177	0.000157	0.000283	-0.000179
r	0.000748	0.000008	-0.000236	0.000439	-0.000179	0.000000

```
> b
```

```
[1] 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.005
```

解方程后，市场组合的结果如下。

```
> w <- solve(Q, b)
```

```
> w <- head(w, -3)
```

```
> w / sum(w)
```

	AAPL	GOOG	MSFT	IBM	T
	-10.154891	4.990912	12.347784	-18.010579	11.826774

2.5 协方差矩阵中的噪声

当我们优化投资组合时，其实我们并没有真实的协方差矩阵和预期收益率向量（它们是均值-方差模型的输入量）。我们使用观测来估计它们，并得到 Q 、 r ，而模型的输出仍然是随机变量。

如果不深入细节，我们可以说模型中会产生惊人的巨大不确定性。尽管有强大数定律的保证，最优的投资组合权重会不时地在 $\pm 200\%$ 之间变动。幸运的

是,如果我们掌握有几年的数据(日收益率),测量风险的相对误差就仅为 20%~25%。

2.6 如果方差不够用

方差可以很容易地度量风险,但也存在一些缺陷。例如,在使用方差时,收益率中的正向变化也会被视为风险的增加。因此,人们开发了一些更复杂的风险度量方法。

比如下面的这个简例,主要关于多种方法运用于之前描述过的 IT_return 资产,它的目的是快速回顾 fPortfolio 包提供的选择。

```
> Spec <- portfolioSpec()
> setSolver(Spec) <- "solveRshortExact"
> setTargetReturn(Spec) <- mean(colMeans(IT_return))
> efficientPortfolio(IT_return, Spec, 'Short')
> minvariancePortfolio(IT_return, Spec, 'Short')
> minriskPortfolio(IT_return, Spec)
> maxreturnPortfolio(IT_return, Spec)
```

这些 R 表达式返回了不同的组合权重,它们的计算方法很多,但没有在这个介绍性的章节中讨论。细节请参考这个包的捆绑文档,比如?portfolio,以及列在参考文献中的相关文章和图书的章节。

2.7 小结

本章的主题是投资组合优化。我们先介绍了主要的思想,接着介绍了马可维茨模型和它的数学公式。我们讨论了可能的求解方法,并运用一个简单的算法来展示这些方法如何运用于真实数据。我们还使用预先写好的 R 包来解决相

第3章 资产定价模型

本章讲述绝对定价的问题 [Cochrane (2005)], 以及如何对支付不确定的资产基于自身的风险确定价值。第2章“投资组合优化”, 在均值方差框架下对个体投资者基于资产收益率分析的决策过程进行建模。本章关注金融市场中是否存在均衡, 均衡需要怎样的条件, 以及均衡如何刻画。我们展示两种主要的方法——资本资产定价模型和套利定价理论, 它们的假设和论证完全不同, 但它们的收益率变动的描述基本相似。

根据相对定价的概念, 基础资产的风险已经包含在自身的价格中, 因此它在衍生品定价中不再发挥深入的作用。这一点会在第6章“衍生品定价”中再次介绍。无套利的观点会要求衍生品价格和基础产品价格保持一致。

本章的主题是展示资产收益率和风险因子之间的关系。我们将会解释如何从若干来源下载数据和清洗数据。使用线性回归来度量相依性, 相关的假设检验刻画了结果的显著性。我们通过一个两步回归的过程检验了单因素指数模型, 并给出了结果的金融学解释。

3.1 资本资产定价模型

第一种类型的模型从经济学的角度解释资产价格。资本资产定价模型 (Capital

Asset Pricing Model, CAPM) 借鉴了上一章中给出的投资组合选择的结果, 解决了以下问题, 一个是通过加总了理性投资者的决策得到了市场总体的看法, 另一个是找出了均衡发展需要的条件。夏普 [Sharpe (1964)] 和林特纳 [Lintner (1965)] 根据下面的假设证明了均衡的存在性。

- 个体投资者是价格的接受者。
- 单阶段的投资界限。
- 投资限定于可交易的金融资产。
- 没有税收, 也没有交易成本。
- 信息免费, 并且所有的投资者都可以获得。
- 投资者是理性的, 遵照均值方差最优化的原则进行决策。
- 期望同质。

假设在一个世界中, 以上这些假设全部成立, 所有的投资者都持有相同的风险资产组合。市场组合包含了所有的证券, 并且每个证券的比例都是该证券的市值占全部证券市值的百分比。市场的风险溢价取决于所有市场参与者的平均风险厌恶程度。得到的均衡结果人们都很熟悉, 就是市场风险溢价与个体证券风险之间的线性关系。

$$E(r_i) - r_f = \beta_i [E(r_m) - r_f] \quad (1)$$

$E(r_i)$ 是某一确定证券的预期收益率, r_f 是无风险收益率, $E(r_m)$ 是市场组合的预期收益率。贝塔 β_i 度量了 CAPM 的风险, 它是这个证券与市场组合的协方差与市场组合收益率方差的函数。

$$\beta_i = \frac{\text{Cov}_{i,m}}{\text{Var}_m} \quad (2)$$

$\text{Cov}_{i,m}$ 是给定证券的收益率和市场组合收益率之间的协方差, Var_m 是市场组合收益率的方差。

贝塔有多种解释。一方面, 贝塔显示了一个股票的收益率相对于市场组合收益率的敏感性, 另一方面, 某个证券的贝塔还显示了该证券向市场组合增加了多少风险。CAPM 说明了, 市场仅仅在系统风险更高的情况下对股票才会给出更高的收益率, 因为非系统风险可以分散, 所以不会为它支付风险溢价。

如果我们重新调整方程 (1) 中的项, 可以得到所谓证券市场线 [Security Market Line (SML)] 的线性方程:

$$E(r_i) = r_f + \beta_i [E(r_m) - r_f] \quad (3)$$

CAPM 说明了, 当市场均衡时, 每个证券都应该位于 SML 上。因此, 这个方程对每个证券或投资组合都成立, 即使在它们都无效的条件下。如果这个方程不满足, 市场就达不到均衡。比如, 如果一个证券的收益率高于它基于 CAPM 计算出的水平, 每个投资者都需要改变他/她的投资组合的组成, 以此降低该证券的收益率并使它满足上面的方程。

3.2 套利定价理论

罗斯 [Ross (1977)] 的套利定价理论 [Arbitrage Pricing Theory (APT)] 也在金融学中用于确定不同证券的收益率。APT 说明, 均衡时不存在套利机会, 并且一个资产的收益率是多个随机因素的线性组合 (Wilmott 2007)。这些因素可以是多种宏观经济因素, 也可以是市场指数。在这个模型中, 每个因素都对应一个特定的贝塔系数。

$$r_i = \alpha_i + \sum_{j=1}^n \beta_{ij} F_j + e_i \quad (4)$$

α_i 是举例证券的常数, β_{ij} 是证券 i 对因素 j 的敏感程度, F_j 是一种系统性因素, 而 e_i 是证券的非系统性风险, 均值为 0。

APT 的核心概念是因素组合 (factor portfolio)。一个因素组合是一个充分分

散的投资组合,它仅仅对一个因素有反应,对所有其他因素的贝塔都为0,对这个特定因素的贝塔为1。假设存在因素组合,使用套利观点可以得到,任何充分分散的投资组合的风险溢价,等于因素组合的风险溢价的加权和(Medvegyev-Száz 2010)。如果这对每一个充分分散的投资组合都成立,单个证券的预期收益率可以通过因素的风险溢价(RP_j)建立以及对这个因素的敏感性 β_{ij} 组合而成。

$$E(r_i) = r_f + \sum_{j=1}^n \beta_{ij} RP_j \quad (5)$$

假如APT模型中仅有一个因素,是市场组合的收益率,那我们把这个模型叫作指数模型。更进一步,如果 α_i 是0,我们会得到CAPM的精确定价公式。

CAPM与APT之间的差异如下。

- CAPM是一个均衡模型,建立于经济学的思考,而APT是一个统计模型,使用了套利观点。
- 在APT理论中,如果投资者有一个充分分散的投资组合,那么可以在实际中通过持有大量资产来建立这个组合,进而可以给出预期收益率——贝塔关系。而CAPM则不能建立所谓的市场组合。
- CAPM说明,每个证券都成立预期收益率-贝塔关系,而APT说明,几乎每个证券都成立这种关系。
- 如果市场中存在错误定价,APT认为,只要有一部分投资者改变投资组合结构就可以,而CAPM认为,每个投资者都会这么做。

3.3 贝塔估计

证券对因子的敏感性可以通过价格的历史运动进行估计。我们将从单因素指数模型中估计贝塔。首先,我们展示处理不同来源的数据以及同步的过程,来显示从不同来源收集并同步数据的处理,然后给出简单的贝塔估计方法,最后建立一个线

性回归模型。

3.3.1 数据选择

我们从 Quandl 下载某个给定股票，比如 Google 价格的时间序列数据，并且下载市场指数价格的时间序列数据，比如 S&P500，时间从 2009 年 6 月 1 日~2013 年 6 月 1 日，正如我们在第 2 章中讨论过的。

```
> library(Quandl)
> Quandl.auth("yourauthenticationtoken")
> G <- Quandl('GOOG/NASDAQ_GOOG',
+   start_date = '2009-06-01', end_date = '2013-06-01')
```

得到的结果 G 是一个含有 6 个分量的变量，对于这些分量，我们只需要收盘价 Close。

```
> str(G)
'data.frame':   1018 obs. of 6 variables:
 $ Date   : Date, format: "2009-06-01" "2009-06-02" ...
 $ Open   : num 419 426 426 435 445 ...
 $ High   : num 430 430 432 441 447 ...
 $ Low    : num 419 423 424 434 439 ...
 $ Close  : num 427 428 432 440 444 ...
 $ Volume : num 3323431 2626012 3535593 3639434 3681002 ...
> G <- G$Close
```

同样的代码运行于 S&P500，尽管我们现在处理的是调整后的收盘价 Adjusted Close。

```
> SP500 <- Quandl('YAHOO/INDEX_GSPC',
+   start_date = '2009-06-01', end_date = '2013-06-01')
> SP500 <- SP500$'Adjusted Close'
```

之所以使用调整的收盘价是因为它们已经根据分红和拆股进行了修正。由于

Google 在这个时期既未支付过任何红利又未曾拆股, 所以不需要在这个例子中调整价格。我们还需要无风险收益率的时间序列, 一个月的美元 LIBOR 利率。尽管我们要处理的是日收益率, 一个月的利率可以视为短期利率, 它比隔夜利率更少地受到随机噪声的影响。

```
> LIBOR <- Quandl('FED/RILSPDEPM01_N_B',
+ start_date = '2009-06-01', end_date = '2013-06-01')
> LIBOR <- LIBOR$Value
```

正如你从前面的 Quandl 调用可以看出, 每次都从不同的数据提供商那里获取数据。这也导致了数据结构上的一些差异, 比如 Google 的 Close 值、S&P500 的 Adjusted Close 值以及的 LIBOR 数据的简单 Values。这些向量的长度看起来都不相等。

```
> sapply(list(G, SP500, LIBOR), length)
[1] 1018 1008 1024
```

这意味着一些时间序列会包括其他序列忽略的日期。我们来定义日期的 intersect 函数并仅仅对那些重新下载之后的单元格结果进行过滤。

```
> G <- Quandl('GOOG/NASDAQ_GOOG',
+ start_date = '2009-06-01', end_date = '2013-06-01')
> SP500 <- Quandl('YAHOO/INDEX_GSPC',
+ start_date = '2009-06-01', end_date = '2013-06-01')
> LIBOR <- Quandl('FED/RILSPDEPM01_N_B',
+ start_date = '2009-06-01', end_date = '2013-06-01')
```

因为 intersect 函数仅能运用于两个向量, 我们调用 Reduce 函数来识别 3 个时间序列中的共同日期。

```
> cdates <- Reduce(intersect, list(G$Date, SP500$Date, LIBOR$Date))
```

现在, 我们简单过滤相关单元格的所有 3 个数据框, 并得到向量。

```
G <- G[G$Date %in% cdates, 'Close']
```

```
SP500 <- SP500[SP500$Date %in% cdates, 'Adjusted Close']
LIBOR <- LIBOR[LIBOR$Date %in% cdates, 'Value']
```

下载和清洗数据之后，你需要使用下面的公式计算股票以及市场指数的对数收益率 (r_{it})。

$$r_{it} = \ln\left(\frac{s_t}{s_{t-1}}\right) \quad (6)$$

s_t 是第 t 日的市场价格。在 R 中，它可以表示成一个函数（细节见第 2 章“投资组合优化”）。

```
> logreturn <- function(x) log(tail(x, -1) / head(x, -1))
```

下一步，应该通过减去无风险的日对数收益率 r_{ft} 来确定风险溢价。因为 LIBOR 利率是基于货币市场基差而报价的，计算时需要遵循（实际天数/360 天）的传统，并且时间序列包含了用百分比表示的利率，因此需要用到下面的公式。

$$r_{ft} = \ln\left(1 + \frac{USDLIBOR}{36000} \times ((t+1) - t)\right) \quad (7)$$

$t+1$ 和 t 都指日期，因此它们的差异是两个收盘值之间的天数，通常为 1，在我们的案例里，如果它们之间没有工作日的话天数会更多。使用下面的命令，结果可以容易地在 R 中计算出来。

```
> rft <- log(1 + head(LIBOR, -1)/36000 * diff(cdates))
> str(rft)
num [1:1001] 1.81e-05 1.81e-05 1.81e-05 1.81e-05 5.42e-05 ...
```

通过计算共同日期的 `diff`，我们已经计算了 $(t+1) - t$ 。并且风险溢价 R_{it} 由下式给出。

$$R_{it} = r_{it} - r_{ft} \quad (8)$$

3.3.2 简单贝塔估计

一旦我们有了这两个时间序列，单个资产（在我们的例子里，是 Google）以及市场（S&P 500）风险溢价，贝塔可以根据方程（2）计算出来。

```
> cov(logreturn(G) - rft, logreturn(SP500) - rft) /
+   var(logreturn(SP500) - rft)
[1] 0.8997941
```

这个计算还可以通过加入一个描述风险溢价的新函数得到简化。

```
> riskpremium <- function(x) logreturn(x) - rft
> cov(riskpremium(G), riskpremium(SP500)) / var(riskpremium(SP500))
[1] 0.8997941
```

这种计算贝塔的方法因为使用风险溢价取代了收益率，所以不同于方程（2）。因为 CAPM 和 APT 都是单期模型，方程两端使用无风险收益率进行修正后并不影响结果。在另一方面，从时间序列估计贝塔时，我们必须决定是否在模型中使用收益率或风险溢价，因为除了在常数无风险收益率的情形下无影响，这两个参数都会有差别（Medvegyev-Száz 2010）。就像我们借鉴金融文献一样，我们复制之前描述过的方法，但是我们必须增加 Merryl Lynch 从收益率中计算的贝塔。

3.3.3 基于线性回归估计贝塔

为了估计贝塔，我们可以使用线性回归，其中的解释变量是市场风险溢价 [(Market Risk Premium (MRP))], 因变量是证券的风险溢价。因此，回归方程的形式如下，证券特征线 [Security Characteristic Line (SCL)] 的公式如下。

$$R_i = \alpha_i + \beta_i R_m + e_i \quad (9)$$

我们将使用普通最小二乘估计 [Ordinary Least Squared (OLS)] 来确定方程 (8) 的线性回归模型。特征线的截距是 α ，股票的收益率中不能被市场因素解释的部分。函数 [方程 (8)] 的斜率表示对市场因子的敏感性，通过贝塔来度量。

使用内置在 R 中的命令 `lm`，我们可以容易地计算出回归模型。

```
> (fit <- lm(riskpremium(G) ~ riskpremium(SP500)))
```

Call:

```
lm(formula = riskpremium(G) ~ riskpremium(SP500))
```

Coefficients:

```
(Intercept) riskpremium(SP500)
0.0002078      0.8997941
```

我们额外增加了圆括号，因此不仅保存了结果还把它们打印了出来。借助于模型，还很容易在图上画出 Google 的特征线，它作为市场风险溢价的函数表示了 Google 的风险溢价。

```
> plot(riskpremium(SP500), riskpremium(G))
> abline(fit, col = 'red')
```

图 3-1 显示了这个结果。 x 轴表示 MRP，而 y 轴表示 Google 的风险溢价。

根据 CAPM， α 等于 0，因此我们假设 α_i 是 0，接着放松这个约束。我们通过在模型中放入 -1，可以使 α 为 0。

```
> fit <- lm(riskpremium(G) ~ -1 + riskpremium(SP500))
```

R 中的回归模型的结果概括如下。

```
> summary(fit)
```

Call:

```
lm(formula = riskpremium(G) ~ -1 + riskpremium(SP500))
```

Residuals:


```

      Min      1Q      Median      3Q      Max
-0.089794 -0.005553 0.000166 0.005520 0.117087

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
riskpremium(SP500) 0.90048    0.03501   25.72 <2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0124 on 1000 degrees of freedom
Multiple R-squared:  0.3982,    Adjusted R-squared:  0.3976
F-statistic: 661.6 on 1 and 1000 DF, p-value: < 2.2e-16

```

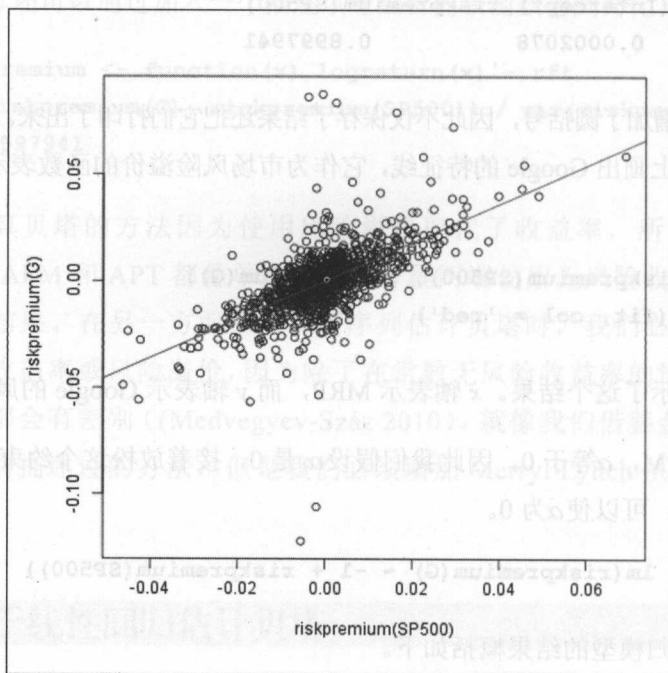


图 3-1 Google 的风险溢价

高的 F-statistic 值说明模型具备解释能力，贝塔证明是显著的，并且，贝塔应该为零的原假设在任何显著性水平上都被拒绝。这些结果与 CAPM 一致。

如果我们通过放松 α 为零的假设来运行检验，可以看到截距与零没有显著的

区别。高的 p 值 p -value 表明不能在任何常见的显著性水平上（高于 90%）拒绝原假设。

```
> summary(lm(riskpremium(G) ~ riskpremium(SP500)))
```

Call:

```
lm(formula = riskpremium(G) ~ riskpremium(SP500))
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.089999	-0.005757	-0.000045	0.005307	0.116883

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.0002078	0.0003924	0.529	0.597
riskpremium(SP500)	0.8997941	0.0350463	25.674	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01241 on 999 degrees of freedom

Multiple R-squared: 0.3975, Adjusted R-squared: 0.3969

F-statistic: 659.2 on 1 and 999 DF, p-value: < 2.2e-16

我们可以在图 3-2 的联合多图上检查残差。

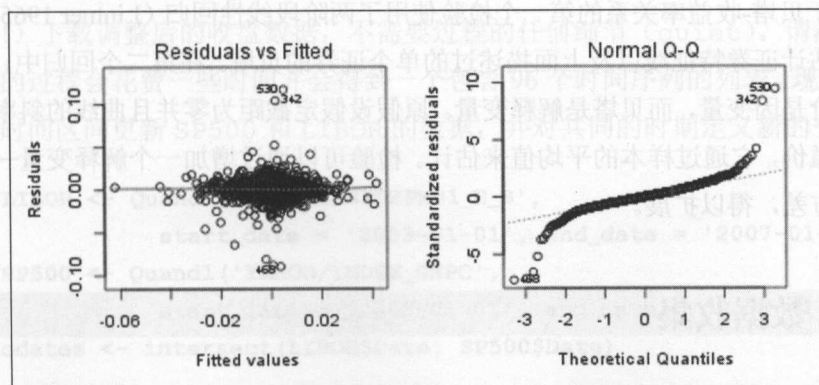


图 3-2 残差的检验

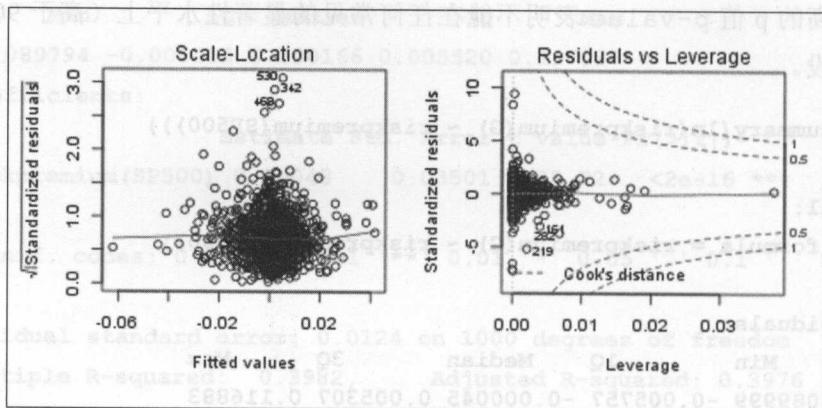


图 3-2 残差的检验 (续)

```
> par(mfrow = c(2, 2))
> plot(fit)
```

你还可以发现 PerformanceAnalytics 包中的内置函数 CAPM.alpha 和 CAPM.beta, 它们可以对一个给定的资产计算阿尔法和贝塔。需要的参数是资产和基准资产的收益率以及无风险利率的序列。

3.4 模型检验

关于贝塔-收益率关系的第一个检验使用了两阶段线性回归 (Lintner 1965)。第一个回归估计证券特征线以及上面描述过的单个证券的贝塔。在第二个回归中, 证券的风险溢价是因变量, 而贝塔是解释变量。原假设假定截距为零并且曲线的斜率为市场的风险溢价, 它通过样本的平均值来估计。检验可以通过增加一个解释变量——单个证券的方差, 得以扩展。

3.4.1 数据收集

我们将使用危机前期 2003 年~2007 年的样本给出检验。因为日数据中的短期

效应更多，所以我们会先基于日时间序列计算月收益率，再在月度数据上运行这个检验。我们下载 2003 年~2007 年按字母表排序的 S&P500 的前 100 个股票价格。

```
> symbols <- c("A", "AA", "AAPL", "ABC", "ABT", "ACE", "ACN", "ACT",
"ADBE", "ADI", "ADM", "ADP", "ADSK", "AEE", "AEP", "AES", "AET", "AFL",
"AGN", "AIG", "AIV", "AIZ", "AKAM", "ALL", "ALTR", "ALXN", "AMAT", "AMD",
"AMGN", "AMP", "AMT", "AMZN", "AN", "ANF", "AON", "APA", "APC", "APD",
"APH", "APOL", "ARG", "ATI", "AVB", "AVP", "AVY", "AXP", "AZO", "BA",
"BAC", "BAX", "BBBY", "BBT", "BBY", "BCR", "BDX", "BEAM", "BEN", "BF.B",
"BHI", "BIIB", "BK", "BLK", "BLL", "BMC", "BMS", "BMY", "BRCM", "BRK.B",
"BSX", "BTU", "BXP", "C", "CA", "CAG", "CAH", "CAM", "CAT", "CB", "CBG",
"CBS", "CCE", "CCI", "CCL", "CELG", "CERN", "CF", "CHK", "CHRW", "CI",
"CINF", "CL", "CLF", "CLX", "CMA", "CMCSA", "CME")
```

请注意，上述列表仅仅包含 96 只股票名，因为其他 4 只股票在参考的时间区间内的缺失值太多。

使用 tseries 包，我们从一个统一的数据库下载这些数据集。

```
> library(tseries)
> res <- lapply(symbols, function(symbol)
+   get.hist.quote(symbol, quote = "AdjClose", quiet = TRUE,
+   start = as.Date('2003-01-01'), end = as.Date('2007-01-01')))
```

因此，我们对每个 symbol 调用 get.hist.quote 函数来从默认的提供商 (Yahoo!) 下载调整后的收盘数据，不需要过程的任何细节 (quiet)。请注意，获取数据的过程会花费一些时间并会得到一个包含 96 个时间序列的列表。现在，我们对新的时间区间更新 SP500 和 LIBOR 的数据，并对共同的时期定义新的交叉。

```
> LIBOR <- Quandl('FED/RILSPDEPM01_N_B',
+   start_date = '2003-01-01', end_date = '2007-01-01')
> SP500 <- Quandl('YAHOO/INDEX_GSPC',
+   start_date = '2003-01-01', end_date = '2007-01-01')
> cdates <- intersect(LIBOR$Date, SP500$Date)
```

如同上面的解释，我们需要月度数据集来替代下载的日度数据。我们选择每个

月的第一个值。为此，我们需要在 Date 格式中保存共同日期的列表。

```
> d <- data.frame(date = as.Date(cdates, origin = '1970-01-01'))
> str(d)
'data.frame': 998 obs. of 1 variable:
 $ date: Date, format: "2003-01-02" "2003-01-03" ...
```

接下来，我们需要把当月的日期和对应的年与月合并起来，三者的数据框完全一样。

```
> d$day <- format(d$date, format = '%d')
> d$my <- format(d$date, format = '%Y-%m')
```

现在，我们对每个 my 组（代表同一年的同一月）的代表当月日期的 day 变量运用 min 函数。

```
> (fds <- with(d, tapply(day, my, min)))
2003-01 2003-02 2003-03 2003-04 2003-05 2003-06 2003-07 2003-08
      "02"      "03"      "03"      "01"      "01"      "02"      "01"      "01"
2003-09 2003-10 2003-11 2003-12 2004-01 2004-02 2004-03 2004-04
      "02"      "01"      "03"      "01"      "02"      "02"      "01"      "01"
2004-05 2004-06 2004-07 2004-08 2004-09 2004-10 2004-11 2004-12
      "03"      "01"      "01"      "02"      "01"      "01"      "01"      "01"
2005-01 2005-02 2005-03 2005-04 2005-05 2005-06 2005-07 2005-08
      "03"      "01"      "01"      "01"      "02"      "01"      "01"      "01"
2005-09 2005-10 2005-11 2005-12 2006-01 2006-02 2006-03 2006-04
      "01"      "03"      "01"      "01"      "03"      "01"      "01"      "03"
2006-05 2006-06 2006-07 2006-08 2006-09 2006-10 2006-11 2006-12
      "01"      "01"      "03"      "01"      "01"      "02"      "01"      "01"
```

我们需要再次把结果和日期合并起来。

```
> (fds <- as.Date(paste(row.names(fds), fds, sep = '-'))))
[1] "2003-01-02" "2003-02-03" "2003-03-03" "2003-04-01" "2003-05-01"
[6] "2003-06-02" "2003-07-01" "2003-08-01" "2003-09-02" "2003-10-01"
[11] "2003-11-03" "2003-12-01" "2004-01-02" "2004-02-02" "2004-03-01"
[16] "2004-04-01" "2004-05-03" "2004-06-01" "2004-07-01" "2004-08-02"
[21] "2004-09-01" "2004-10-01" "2004-11-01" "2004-12-01" "2005-01-03"
```



```
[26] "2005-02-01" "2005-03-01" "2005-04-01" "2005-05-02" "2005-06-01"
[31] "2005-07-01" "2005-08-01" "2005-09-01" "2005-10-03" "2005-11-01"
[36] "2005-12-01" "2006-01-03" "2006-02-01" "2006-03-01" "2006-04-03"
[41] "2006-05-01" "2006-06-01" "2006-07-03" "2006-08-01" "2006-09-01"
[46] "2006-10-02" "2006-11-01" "2006-12-01"
```

并对上述确定的日期再次过滤 res 数据框。

```
> res <- lapply(res, function(x) x[which(zoo::index(x) %in% fds)])
```

接着, 在合并列表和一个时间序列之后, 可以直接把列表转化为列名灵活的常见 data.frame 格式。

```
> res <- do.call(merge, res)
> str(res)
'zoo' series from 2003-01-02 to 2006-12-01
Data: num [1:48, 1:96] 17.8 15.3 12.1 12.5 15 ...
Index: Date[1:48], format: "2003-01-02" "2003-02-03" ...
> res <- as.data.frame(res)
> names(res) <- symbols
```

结果产生一个 48 行 96 列的数据框。我们依然需要在列基础上计算每一个下载股票的收益率。但是, 为了这个目的, rft 需要根据每个月的第一个值来更新。

```
> LIBOR <- LIBOR[LIBOR$Date %in% fds, 'Value']
> rft <- log(1 + head(LIBOR, -1)/36000 * as.numeric(diff(fds)))
> res <- apply(res, 2, riskpremium)
```

我们把 S&P 500 的值作为月数据集来过滤。

```
> SP500 <- SP500[SP500$Date %in% fds, 'Adjusted Close']
```

3.4.2 对 SCL 建模

使用股票收益率的时间序列, 我们能对每个证券计算贝塔。因此, 我们会得到

用样本均值表示的风险溢价向量和一个包含贝塔的向量。

第二个回归的估计如下。

$$\bar{R}_i = \gamma_0 + \gamma_1 \beta_i \quad (10)$$

计算完每只股票的 riskpremium 参数并把相关结果存入一个 data.frame 之后, 可以通过基本循环一次性地计算每只股票的风险 premium 参数以及收益率的均值。

```
> res <- apply(res, 2, riskpremium)
> res <- as.data.frame(res)
> r <- t(sapply(symbols, function(symbol)
+       c(beta = lm(res[, symbol] ~
+                   riskpremium(SP500))$coefficients[[2]],
+       mean = mean(res[, symbol]))
+ ))
> r <- as.data.frame(r)
```

因此, 通过迭代所有的股票代码, 我们来画出贝塔计算值的返回列表以及风险溢价的均值, 如图 3-3 所示。

```
> plot(r$beta, r$mean)
> abline(lm(r$mean ~ r$beta), col = 'red')
```

这个模型可以描述如下。

```
> summary(lm(r$mean ~ r$beta))
```

Call:

```
lm(formula = r$mean ~ r$beta)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.024046	-0.008783	-0.003475	0.006485	0.039731

Coefficients:

Estimate	Std. Error	t value	Pr(> t)

```

(Intercept) 0.009084    0.002429    3.740 0.000325 ***
r$beta      0.005528    0.001678    3.295 0.001413 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01383 on 89 degrees of freedom
(5 observations deleted due to missingness)
Multiple R-squared:  0.1087,    Adjusted R-squared:  0.09873
F-statistic: 10.86 on 1 and 89 DF, p-value: 0.001413

```

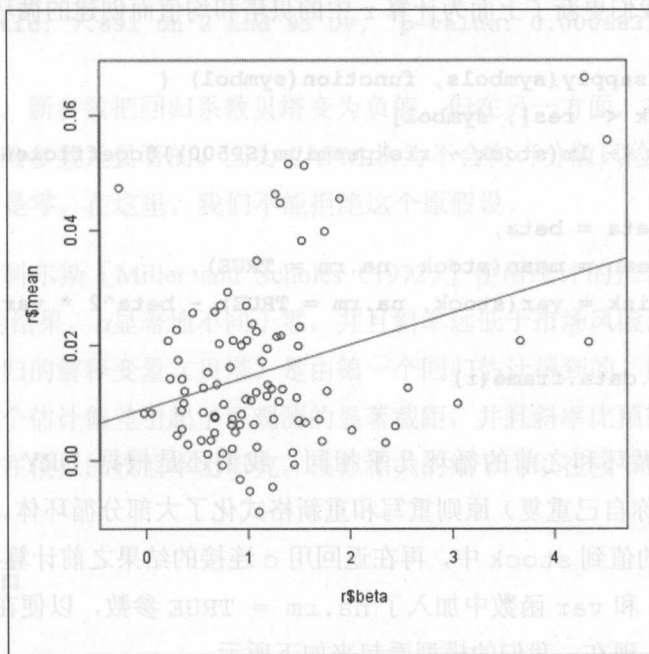


图 3-3 贝塔及风险溢价的均值

根据上面的结果，截距为正，但它和零没有显著的区别。基于月度基础，SML的斜率等于0.5528%，这低于预期，根据原假设，它应该是该时期市场溢价的平均值0.69%。但是，这个差异仍然在统计上显著。根据检验，不能拒绝贝塔-收益率的关系。

3.4.3 检验个体方差的解释能力

检验可以进一步地深化，包括把非系统风险作为第二个解释变量的检验。证券

的个体风险可以如下计算。

$$\sigma_{ei}^2 = \sigma_i^2 - \beta_i^2 \sigma_m^2 \quad (11)$$

因此，我们必须先计算方差的向量，接着得到单个方差的向量。回归方程如下估计。

$$\bar{R}_i = \gamma_0 + \gamma_1 \beta_i + \gamma_2 \sigma_{ei}^2 \quad (12)$$

到现在，我们更新了上面为计算 r 中的贝塔和均值而创建的循环。

```
> r <- t(sapply(symbols, function(symbol) {
+   stock <- res[, symbol]
+   beta <- lm(stock ~ riskpremium(SP500))$coefficients[[2]]
+   c(
+     beta = beta,
+     mean = mean(stock, na.rm = TRUE),
+     risk = var(stock, na.rm = TRUE) - beta^2 * var(SP500))
+ })))
> r <- as.data.frame(r)
```

尽管这个循环和之前的循环几乎相同，我们还是根据 DRY (Don't Repeat Yourself, 别和你自己重复) 原则重写和重新格式化了大部分循环体。因此，我们先存储 `symbol` 的值到 `stock` 中，再在返回用 `c` 连接的结果之前计算了贝塔。现在，我们又在 `mean` 和 `var` 函数中加入了 `na.rm = TRUE` 参数，以便在计算之前移除可能的缺失值。现在，我们的模型看起来如下所示。

```
> summary(lm(r$mean ~ r$beta + r$risk))
```

Call:

```
lm(formula = r$mean ~ r$beta + r$risk)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.023228	-0.009175	-0.003657	0.006817	0.036262

Coefficients:

```

      Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.400e-02  3.711e-03   3.772 0.000285 ***
r$beta      -1.743e-03  4.677e-03  -0.373 0.710293
r$risk      -9.956e-08  5.798e-08  -1.717 0.089266 .
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Residual standard error: 0.01381 on 93 degrees of freedom
Multiple R-squared:  0.1451,    Adjusted R-squared:  0.1267
F-statistic: 7.891 on 2 and 93 DF,  p-value: 0.0006833

```

有趣的是，新参数把回归系数贝塔变为负值。但在另一方面，在 95% 的显著性水平上证明风险参数是显著的。因为 CAPM 认为不会向可分散风险支付风险溢价，原假设假定 β_2 是零。在这里，我们不能拒绝这个原假设。

米勒和斯科尔斯 [Miller and Scholes (1972)] 使用统计的推理解释了第一个 CAPM 检验的结果， α 显著地不同于零，并且斜率远低于市场风险溢价的平均值。因为第二个回归的解释变量（贝塔）是由第一个回归估计得到的，所以它还包含着统计误差。这个估计偏差引起了可观测的显著截距，并且斜率比预期 SML 更平坦。这种观点可以在模拟的收益率上研究。模拟相关的细节可以在接下来的两章找到。

3.5 小结

在本章中，资产收益率的系统风险通过它们对市场方差的贡献（贝塔）来度量。我们使用线性回归来量化这种关系。为了确认资本资产定价模型的观点运行假设检验。

第 4 章

固定收益证券

在第 3 章“资产定价模型”中，我们关注的模型建立在风险与价格的关系之上：风险通过贝塔来度量，价格包括了金融产品和投资组合的价格。第一个模型 CAPM 使用均衡方法，而第二个模型 APT 建立在无套利假设之上。

固定收益投资组合管理的一般目标是在给定的风险/回报配置下建立固定收益证券的投资组合。换句话说，投资组合经理的任务是，坚持给定的投资目标，依照最大化投资组合的预期收益率的原则把资金配置于不同的固定收益证券。

这个过程包括了对收益率曲线、预付行为以及默认证券的动态建模。需要使用的工具是时间序列分析、随机过程和最优化。

固定收益证券的风险包括信用风险、流动性风险以及市场风险等。仅仅选择违约风险预先给定的证券就可以处理前两种风险，比如说，信用评级最低和流动性特征合理。固定收益证券的市场风险通常可以通过久期、修正久期、主要久期或者因子久期来刻画。所有这些利率风险的度量都可以用于固定收益证券。本章关注固定收益证券的市场风险。

4.1 度量固定收益证券的市场风险

给定了收益率曲线，求解出固定收益证券现值的一般公式为：
$$P = \sum_{t=0}^T \frac{CF_t}{(1+y_t)^t},$$

其中, T 是证券到期的时间, CF_t 是证券在时间 t 的现金流, y_t 是投资者在时间 t 收到现金的折现率。随着时间流逝, 债券的市场价格会收敛到它的票面价值。这种价格的改变体现在期望意义上, 它因此没有考虑风险。市场风险源于利率的变动, 它引起了再投资风险和流动性风险。前者会影响到可用于再投资的息票利息支付的收益率, 后者会影响到债券的市场价格。

研究债券价格的到期收益率 y 的函数公式 $P = \sum_{t=0}^T \frac{CF_t}{(1+y)^t}$, 可以度量利率变动的市场价格影响。 $\Delta P = \frac{dP}{dy} \Delta y + \frac{1}{2} \frac{d^2 P}{dy^2} (\Delta y)^2 + \dots$, 在收益率中一个 Δy 的变动引起的价格变动的百分比可以表示为: $\frac{\Delta P}{P} = \frac{1}{P} \frac{dP}{dy} \Delta y + \frac{1}{2P} \frac{d^2 P}{dy^2} (\Delta y)^2 + \dots$, $\frac{\Delta P}{P}$ 的二阶近似是 $\frac{\Delta P}{P} = -D^* \Delta y + \frac{1}{2} \text{Convexity} (\Delta y)^2 = -\frac{D}{1+y} \Delta y + \frac{1}{2} \text{Convexity} (\Delta y)^2$, 当按时期表达收益率的公式时, 复合久期 (D)、修正久期 (D^*) 以及凸度分别定义如下。

$$D = -\frac{1}{P} \frac{dP}{dy} (1+y) = \sum_{t=0}^T t \times \frac{CF_t / (1+y)^t}{P}$$

$$D^* = D / (1+y)$$

$$\text{Convexity} = \frac{1}{P} \frac{d^2 P}{dy^2} = \frac{1}{P} \frac{1}{(1+y)^2} \sum_{t=0}^T \frac{CF_t (t^2 + t)}{(1+y)^t}$$

显然, 债券的定价公式表现了债券的到期收益率 (y) 与价格 (P) 之间的反向关系。因为久期刻画了债券相对于价格变动而产生的到期收益率的变动, 因此它是债券利率风险的最重要度量。久期是债券期限的加权平均。

案例——R 实现

考虑一个到期时支付票面价值 1000 美元的 10 年期债券, 年化息票率为 8%, 利息每个季度支付一次, 同时假定连续复利收益率曲线在 10% 处平坦。

为了计算上面描述的指数, 我们使用 GUIDE 包, 它对多种金融计算和金融衍

生品定价的交互图形提供了图形用户界面,因此下面的案例与其他各章相比,参数设置的方式更加直观。

在安装和载入这个包之后,主程序可以通过 GUIDE 函数来启动。

```
> install.packages('GUIDE')
> library(GUIDE)
> GUIDE()
```

这会载入主窗口,它的菜单可以访问包内的 55 个函数,如图 4-1 所示。

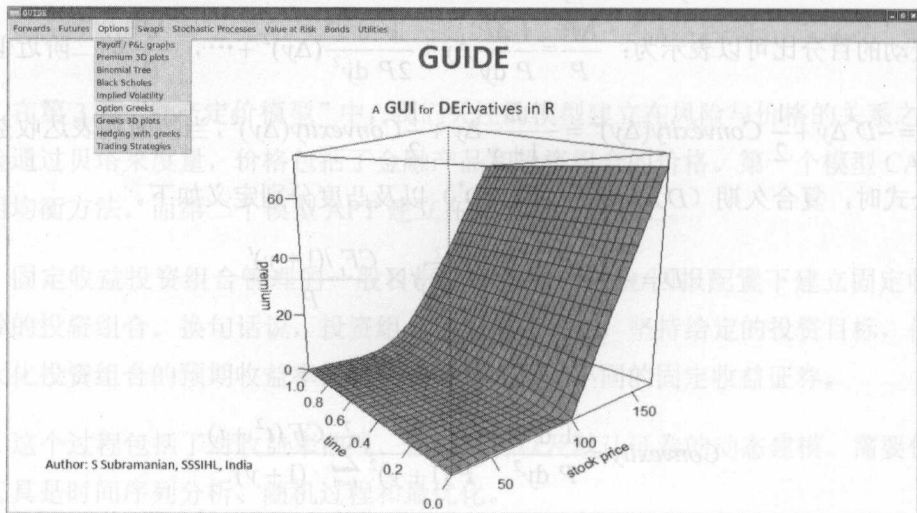


图 4-1 GUIDE 包的主窗口

这个函数可以通过顶部菜单栏旁边的 R 直接命令来调用。

通过 bondprice 函数,可以很快地计算出公允价值为 867.28 美元。priceyield 函数表现了折现率与债券价格之间的反向关系。债券久期根据 bonddur 函数决定,如图 4-2 所示。

这个函数可以通过设置来反映年度或半年度息票支付,并且折现率的频率可以变化。这个函数也考虑到修正久期的计算。同一个函数的凸度是由函数 bondconv 计算的,显示在图 4-3 中。

Bond Duration

Face Value: 1000

- 8 + Coupon (% p.a.)

- 10 + Discount Rate (% p.a.)

- 10 + Maturity (Yrs)

Coupon payments

- ☒ quarterly
- ☐ semi-annual
- ☐ annual

Frequency of discount rate

- ☒ continuous comp
- ☐ same as coupon freq
- ☐ annual comp

Duration formula

- ☒ Macaulay
- ☐ Modified

Duration: 6.72

图 4-2 债券的久期计算

Bond Convexity

Face Value: 1000

- 8 + Coupon (% p.a.)

- 10 + Discount Rate (% p.a.)

- 10 + Maturity (Yrs)

Coupon payments

- ☒ quarterly
- ☐ semi-annual
- ☐ annual

Frequency of discount rate

- ☒ continuous comp
- ☐ same as coupon freq
- ☐ annual comp

Convexity: 57.66

图 4-3 债券的凸度计算

请注意，当折现率表示为连续复利收益率，凸度由下面的公式给出。

$$Convexity = \frac{1}{P} \frac{d^2 P}{dy^2} = \frac{1}{P} \sum_{t=0}^T CF_t \exp(-yt) t^2$$

可以使用 `duryield` 函数和 `durcoupon` 函数来评价增加的收益率如何影响久期，以及更大的息票利息如何冲击债券久期。

久期和到期之间的联系可以由 `durmaturity` 函数显示，如图 4-4 所示。

在算出一只债券的久期和凸度之后，也很容易计算组合中单个元素的久期和凸度的加权平均，当作投资组合的久期和凸度。

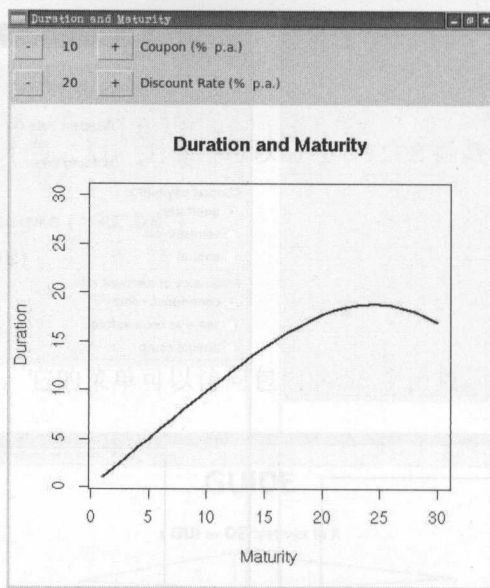


图 4-4 久期和到期之间的关系

其他的包，比如 `maRketSim` 和 `termstrc`，也包含了可以计算债券甚至是完整投资组合的久期、修正久期和凸度的函数。

4.2 固定收益投资组合的免疫

如果一个投资组合不会受到利率改变的影响，那么它是免疫的。久期很好地度量了利率的敏感性，因此，它广泛地应用于免疫的投资组合。因为使用久期假定了收益率曲线形态的平坦和微小的平移，免疫的投资组合受到这些假定的约束，而不受影响意味着投资组合仅仅在收益率变动的时候会发生轻微的改变。

免疫策略有两种不同的类型：净值免疫和目标日期免疫。

4.2.1 净值免疫

固定收益的投资组合经理常常会对收益率曲线的未来变动方向持有某种看法。

我们假定一位投资组合经理预期最近利率会上升。因为这会对投资组合造成不利影响，这位经理可能会决定参与远期协议或利率互换，从而把组合的久期设置为零。这些工具改变了组合的久期，和有助于把组合的久期设置为零，并且还无需清算整个组合。

投资组合经理还有另外一个目的，根据基准投资组合的久期设置组合的久期。这有助于超越基准投资组合的表现，并调整对市场运动的预期。

通常银行更加关心从市场的价格变动中保护他们的权益价值。他们可以通过把权益久期设置为理想水平来实现这个目标。

4.2.2 目标日期免疫

假设有这样一个债务现金流给定的投资者，他可以构建一个固定收益证券的资产组合，使它的久期等于债务的久期，从而可以实现投资组合的免疫。这种目标日期免疫可以确保投资组合中未来的支付义务匹配资产的需求。比如，这种处理可以通过 `genPortfolio.bond` 函数来完成。

4.2.3 定制

定制 (Dedication) 策略是一种类型特殊的目标日期免疫，其中资产的现金流和债务中的每个部分相匹配。为了实现这种免疫，一种方式是通过零息债券为相应的债务部分融资。

4.3 可转换债券的定价

可转换债券通常由低信用评级和高增长潜力的公司发行。这些公司通过赋予债券持有者把债券转换为特定数量的发行公司股票份额的权利 (但是没有义务)，使他

们的利息成本降低。投资者既获得了转换为股权的潜在好处，同时又得到了债券现金流下跌的保护。如果可转债转换了，公司的杠杆会下降，从而公司会获益，但代价是债券转换时股票被稀释。

这些特点说明可转换债券有3个不同阶段：实值可转换债券（转换价格 < 股价）的行为类似股票，平价可转换债券（转换价格 = 股价）可以当作股票和债务，而虚值可转换债券（转换价格 > 股票价格）的行为更像债务。即使在 Black-Scholes-Merton 模型框架内，可转换债券的定价也会很复杂，但基本原则是分别对债券和期权定价。

我们来考虑一个5年期的可转换债券，票面价值100美元，息票率5%，按年付息，可以选择在票面到期时转换为4股普通股。假定无风险收益率对所有的到期都是5%，债券的信用利差是2%，基础股票的价格是20美元，股票的波动率是20%，而股息率是0。可以使用R估计这个可转换债券的价值。首先，我们把将会使用的日期定义为今天。

```
> today <- Sys.Date()
```

接下来，我们设置交易和结算日期，并给定一条平坦的收益率曲线（基于times参数，是一个从现在开始，从0到10步长为0.1的序列），来计算折现曲线的价值。

```
> params <- list(tradeDate = today - 2,
+               settleDate = today,
+               dt = 0.25)
> times <- seq(0, 10, 0.1)
> dividendYield <- DiscountCurve(params, list(flat = 10e-6), times)
> riskFreeRate <- DiscountCurve(params, list(flat = 0.05), times)
```

上面的股息率、无风险利率以及接下来固定基础资产的价格和波动率会稍后输入 Black-Scholes 过程，最后对这个债券初步建立二项式定价。

```
> process <- list(
+   underlying = 20,
+   divYield = dividendYield,
+   rff = riskFreeRate,
```

```
+ volatility = 0.2)
```

我们还需要设定转换比率，它决定了如果债券持有者决定把债券转换为股票，那他会得到多少普通股。同时也在这里设定债券的票面价值和信用利差。

```
> bondparams <- list(
+   exercise      = "eu",
+   faceAmount    = 100,
+   redemption    = 100,
+   creditSpread   = 0.02,
+   conversionRatio = 4,
+   issueDate      = as.Date(today + 2),
+   maturityDate   = as.Date(today + 1825))
```

使用年度息票支付。

```
> dateparams <- list(
+   settlementDays = 3,
+   dayCounter     = "ActualActual",
+   period         = "Annual",
+   businessDayConvention = "Unadjusted")
```

并把上面设定的参数输入 ConvertibleFixedCouponBond 函数。

```
> ConvertibleFixedCouponBond(bondparams, coupon = 0.05, process,
dateparams)
```

Concise summary of valuation for ConvertibleFixedCouponBond

Net present value : 107.1013

clean price : 107.06

dirty price : 107.1

accrued coupon : 0.041096

yield : 0.033848

cash flows :

Date	Amount
2014-06-21	4.9589
2015-06-21	5.0000
2016-06-21	5.0073
2017-06-21	4.9927

2018-06-21 5.0000

2018-06-21 100.0000

如果排除了可转换特征, 债券价值大约为 92 美元, 而带有额外特征的债券价值变成 107.1 美元。现在, 我们从 1~30 提高基础股票的价格, 同时检查净现值的改变。

```
> res <- sapply(seq(1, 30, 1), function(s) {
+   process$underlying = s
+   ConvertibleFixedCouponBond(bondparams, coupon = 0.05, process,
+   dateparams)$NPV
+ })
> plot(1:30, res, type = 'l',
+   xlab = 'Price of the underlying stocks',
+   ylab = 'Net Present Value')
```

图 4-5 显示了基础股票价格和可转换债券的计算价值之间的变动关系。

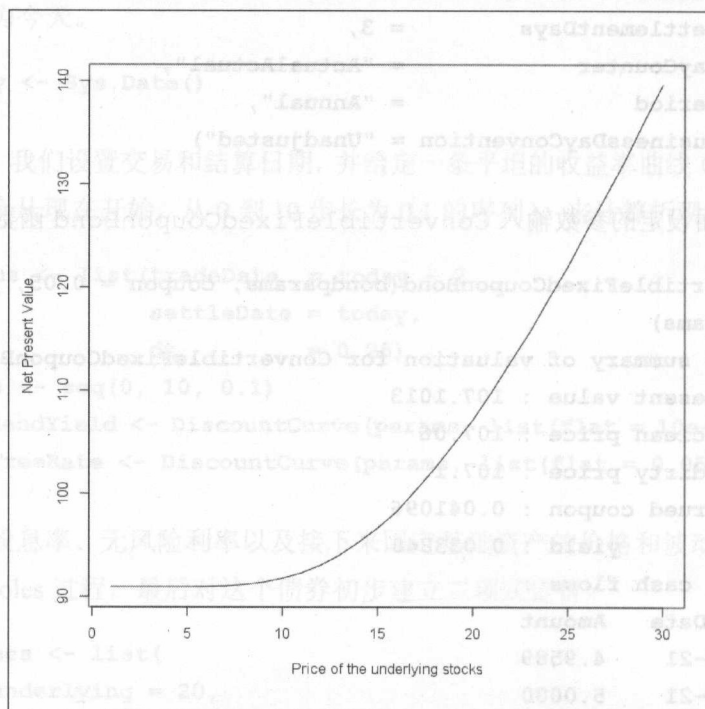


图 4-5 基础股票的价格和可转换债券的净现值

4.4 小结

在本章中，我们使用 R 来度量一个固定收益投资组合的风险。我们用到了 `GUIDE` 包选择的函数，并运用了 `RQuantLib` 包的可转换债券的定价函数。在下一章，你将学到如何使用 R 来估计即期收益率曲线。

5.2 估计问题

想象有 n 只债券交易，它们的毛价格（或脏价格）表示为 $p \in \mathbb{R}^n$ 。有 m 个时期，其中至少一位债券所有者会收到现金流。在时期 t 的现金流记为 $c_t \in \mathbb{R}^n$ ，其中 $0 \leq t \leq T$ 。 $n \times m$ 的矩阵 C 包含债券的现金流。我们把债券价格作为债券现金流的现值与正态分布的误差项之和来建模。

假设 $p = [p_1, \dots, p_n]'$ 是 $n \times 1$ 的向量，表示 n 只债券的毛价格。假设 $c_t = [c_{t1}, \dots, c_{tn}]'$ 是 $n \times 1$ 的向量，表示在时期 t 的现金流。假设 $C = [c_1, \dots, c_m]$ 是 $n \times m$ 的矩阵，表示债券的现金流。假设 $\epsilon_t = [\epsilon_{t1}, \dots, \epsilon_{tn}]'$ 是 $n \times 1$ 的向量，表示在时期 t 的误差项。假设 ϵ_t 服从正态分布，均值为 0，方差为 $\sigma^2 I$ ，其中 I 是 $n \times n$ 的单位矩阵。假设 ϵ_t 与 ϵ_s 独立，对于 $t \neq s$ 。假设 ϵ_t 与 p 独立。假设 p 服从正态分布，均值为 μ ，方差为 Σ 。假设 $\mu = [0, \dots, 0]'$ 。假设 $\Sigma = \sigma^2 I$ 。假设 $\sigma^2 = 1$ 。假设 I 是 $n \times n$ 的单位矩阵。假设 ϵ_t 服从正态分布，均值为 0，方差为 $\sigma^2 I$ ，其中 I 是 $n \times n$ 的单位矩阵。假设 ϵ_t 与 ϵ_s 独立，对于 $t \neq s$ 。假设 ϵ_t 与 p 独立。假设 p 服从正态分布，均值为 μ ，方差为 Σ 。假设 $\mu = [0, \dots, 0]'$ 。假设 $\Sigma = \sigma^2 I$ 。假设 $\sigma^2 = 1$ 。假设 I 是 $n \times n$ 的单位矩阵。

$$\left(\frac{1}{(1+b)^t} \right) \frac{d}{dt} = (1+b)^{-t}$$

这里的 b 是在交易双方基于假设的远期贷款协议商定的利率，当这个合约

在本章中，我们使用 R 来度量一个固定收益投资组合的风险。我们用到了 `GUIDE` 包选择的函数，并运用了 `RQuantLib` 包的可转换债券的定价函数。在下一章，你将学到如何使用 R 来估计即期收益率曲线。

2018-06-21 5.0000

2018-06-21 100.0000

第5章

如果排除了可转换特征，债券价值大约为 92 美元，而带有额外特征的债券价值

第5章 利率期限结构

第5章 利率期限结构

```
+ processUnderlying =  
+ ConvertibleFixedCoupon  
dateparams)$NPV  
+ i)  
> plot(1:30, res, type = 'l',  
+ xlab = 'Price of the underlying stocks',
```

第 5 章 估计利率期限结构

在前一章中，我们讨论了利率水平的变动，即期限结构如何影响了固定收益证券的价格。现在，我们关注利率期限结构的估计，这是金融学的一个基本概念。在几乎所有的金融决策中，它都是一个重要的依据。本章将介绍通过三次样条回归的方法估计期限结构，并会演示如何使用 `termstrc` 包和 `govbonds` 包。

5.1 利率期限结构与相关函数

一个面值 1 美元的 t -年期零息债券，是一种在到期时刻，即 t 年时支付 1 美元的证券。令 $d(t)$ 表示它的市场价值，也叫作 t 年折现因子。函数 $d:[0,T] \rightarrow R$ 叫作折现函数。无套利假设通常假定 $d(0)=1$ 和 $d(t)$ 单调下降，并且 $d(t)>0$ 。还有通常假设 $d(t)$ 二阶连续可微。

令 $r(t)$ 是 t -年零息债券的连续复合年化收益率，它的定义如下。

$$r(t) = \frac{1}{t} \ln \left(\frac{1}{d(t)} \right)$$

函数 $r:[0,T] \rightarrow R$ 叫作（零息债券）收益率曲线。

令 $f:[0,T] \rightarrow R$ 表示瞬时远期利率曲线或简称为远期利率曲线，其中，

$$f(t) = \lim_{h \rightarrow 0} \frac{\ln |d(t)/d(t+h)|}{h} = -\frac{d'(t)}{d(t)}$$

在这里, $f(t)$ 是在交易双方基于假设的远期贷款协议商定的利率, 当这个合约签订时, 其中的一方承诺在 t 年的时刻以一个非常短的期限和这个谈好的固定利率, 向另一方贷出某个数目的资金。

折现函数、收益率曲线以及远期利率曲线彼此相互依赖, 它们都可以是利率期限结构的某种表达形式。期限结构可以和其中的任何函数或者全部函数有关。

5.2 估计问题

我们不能直接观测期限结构, 但是我们能观测依赖于价格期限结构的产品的市场价格, 并由此估计期限结构。对于期限结构的一个好的信息来源是政府债券市场, 其中交易着大量的价格仅仅依赖于期限结构的流动性证券。

想象有 n 只债券交易, 它们的毛价格 (或肮脏价格) 表示为 $p \in R^n$ 。有 m 个日期, 其中至少一位债券所有者会收到支付。这些支付分别预定于 t_1, t_2, \dots, t_m 年时刻收取, 其中 $0 < t_1 < \dots < t_m = T$ 。 $n \times m$ 的矩阵 C 包含债券的现金流。我们把债券价格作为债券现金流的现值与正态分布的误差项之和来建模。

$$p = Cd + \varepsilon \quad (1)$$

这里 d 是一个包含折现因子 $d(t_j)$ 的向量, ε 是一个包含误差项的向量。出于两种可能的理由, 一个是债券的观测市场价格中存在观测误差, 一个是市场中存在着轻微的不完美, 比如交易成本会使理论价格与市场价格之间存在差异, 并且这种差异无法用于套利交易, 从而使债券的观测市场价格不等于债券现金流的现值。不同债券的误差方差也有差异。

$$E(\varepsilon) = 0$$

$$E(\varepsilon \varepsilon') = \sigma^2 \Omega$$

这里的 Ω 是一个 $n \times n$ 半正定的对角阵。假设债券价格中的误差标准差与它的买卖价

差（即，债券的买入报出价格和卖出报出价格的差异）成比例，这是符合逻辑的。于是， ω_{ii} 通常选取为证券 i 报价差的平方。

看起来，公式（1）是一个典型的线性回归，但通常来说它无法直接估计，因为观测（债券价格）的个数常常少于待估计的系数个数。因此，我们建模时需要待估计的参数个数减少的期限结构模型，并需要保证期限结构的估计结果合理。

5.3 基于线性回归的期限结构估计

假设折现函数可以表达为二次连续可微的函数 f_1, f_2, \dots, f_l 的线性组合。

$$d(t) = \sum_{k=1}^l w_k f_k(t)$$

其中，

$$d(0) = \sum_{k=1}^l w_k f_k(0) = 1$$

通过广义最小二乘法，我们可以估计权重 w_k 。随后我们会讨论函数 f_k 的选择。需要估计的折现函数是估计权重 \hat{w}_k 的函数。

$$\hat{d}(t) = \sum_{k=1}^l \hat{w}_k f_k(t)$$

令 D 表示 $m \times l$ 矩阵，它的 d_{jk} 元素是 $f_k(t_j)$ ，并且 $w \in R^l$ 是包含权重 w_k 的向量。于是， $d = Dw$ ，并且，

$$p = CDw + \varepsilon \quad (2)$$

这是一个在约束 $d(0)=1$ 之下的线性回归模型，该约束也可以表示如下。

$$r'w = 1 \quad (3)$$

其中 $r' = (f_1(0), f_2(0), \dots, f_l(0))$ 。

在约束方程（3）之下，方程（2）的权重的广义最小二乘估计如下。

$$\hat{w} = w^* - (X'\Omega^{-1}X)^{-1}r[r'(X'\Omega^{-1}X)^{-1}r]^{-1}(r'w^* - 1)$$

其中, $X=CD$, $w^*=(X'\Omega^{-1}X)^{-1}X'\Omega^{-1}p$ 。

5.4 三次样条回归

如果我们希望估计可以得到一个合理的折现函数结果, 那么需要仔细选择函数 f_k 。典型的折现函数是非线性的。这是一个单调递减函数, 在无穷远处渐进收敛到 0。因此, 拟合一条直线不是好主意。可以尝试拟合一个多项式折现函数, 不过这个解也不令人满意。如果我们拟合低阶多项式, 它们通常不够灵活, 不会很好地拟合, 特别是在短期到期日。如果我们拟合高阶多项式, 它们可以很好地拟合, 但债券相对较少时会在债券的长期到期期限上产生剧烈波动。这些剧烈波动常常会产生不符合实际的期限结构估计结果。

样条函数可以帮助解决上述问题, 它能够在不增加估计函数的多项式阶数的前提下, 在需要的局部增加灵活性。最早 McCulloch 在 1971 年提出, 通过对折现函数拟合三次样条来估计期限结构。

三次样条是实函数, 定义域是实数轴上的一个区间。定义域 $[b_0, b_K]$ 通过所谓的节点 (knot points) b_0, b_1, \dots, b_K 划分为子区间, 其中 $b_0 < b_1 < \dots < b_K$ 。在每个子区间上, 三次样条函数是一个三次多项式。这些三次多项式在节点处连接起来, 这样样条函数是连续的, 并且它在 $[t, T]$ 上是二阶连续可微的。每一个 $[0, T]$ 上的三次样条函数以及一个给定的节点集合 b_0, b_1, \dots, b_K 可以表达为 $K+3$ 个样条基函数的线性组合, 样条基函数是覆盖在同样节点上的三次样条函数。因此, 如果想对折现函数拟合一个三次样条, 我们可以简单地把函数 f_k 选择为三次基样条, 接下来会阐述在 govbonds 数据集的德国政府债券数据中。

```
> data(govbonds)
> str(govbonds[['GERMANY']])
List of 8
 $ ISIN      : chr [1:52] "DE0001141414" "DE0001137131" "DE0001141422"
               "DE0001137149" ...
 $ MATURITYDATE: Date[1:52], format: "2008-02-15" "2008-03-14" ...
 $ ISSUEDATE   : Date[1:52], format: "2002-08-14" "2006-03-08" ...
```

```

$ COUPONRATE : num [1:52] 0.0425 0.03 0.03 0.0325 0.0413 ...
$ PRICE      : num [1:52] 100 99.9 99.8 99.8 100.1 ...
$ ACCRUED    : num [1:52] 4.09 2.66 2.43 2.07 2.39 ...
$ CASHFLOWS  : List of 3
  ..$ ISIN   : chr [1:384] "DE0001141414" "DE0001137131" "DE0001141422"
  "DE0001137149" ...
  ..$ CF     : num [1:384] 104 103 103 103 104 ...
  ..$ DATE   : Date[1:384], format: "2008-02-15" "2008-03-14" ...
$ TODAY      : Date[1:1], format: "2008-01-30"

```

这个数据集包含 52 只德国债券，其中我们会关心它们的发行日期和到期日期、价格，以及提供的现金流。为了深入分析，接下来需要创建一个类似的数据集，请参考 `govbonds` 的示例。

首先，我们使用 `prepro_bond` 函数预处理债券数据集，返回现金流、期限、到期收益率、基于久期的权重矩阵加上肮脏价格以及累计利息向量，等等。

```
> prepro <- prepro_bond('GERMANY', govbonds)
```

一个重要的问题是设定节点的个数以及放置节点。第一个节点和最后一个节点分别是 0 和 T ，其他节点的选择需要使每个子区间内的到期债券的数量大致相同。设定节点的个数就不那么直接。这将会决定要估计的参数个数，更加会影响待估计的期限结构。我们可以通过设置 $K=1$ 开始估计过程，接着循环增加一个节点并进行估计，直到拟合优度明显提高并且估计的期限结构表现良好为止。或者，也可以遵循 McCulloch 提出的经验法则：节点的个数近似为 \sqrt{n} 。现在，借助于从 `prepro` 对象分解出的到期日矩阵，我们在下面的命令中说明这个过程。

```
> m <- prepro$m[[1]]
```

我们用 n 表示德国债券的个数， s 表示它的取整的平方根（结点的个数）， s 的结果是 7。

```
> n <- ncol(m)
```

```
> s <- round(sqrt(n))
```


如果 s 的计算结果小于 3, 那么识别理想的节点很容易。我们将使用下面的命令。

```
> c(floor(min(y[, 1])), max(m[, ncol(m)]))
```

在这里, 我们对结果进行向下取整 (floor) 得到期限矩阵, 并从矩阵中识别第一列中最小 (min) 的数, 以及最后一列 (ncol) 中最大 (max) 的数。

在这里, 如果 s 大于 3, 那么按照上面的命令行来定义第一个节点和最后一个节点, 再使用一些长度为 $s-3$ 的辅助向量来计算两者之间的其他节点, 如下所示。

```
> i <- 2:(s-2)
> h <- trunc(((i - 1) * n) / (s - 2))
> theta <- ((i - 1) * n) / (s - 2) - h
```

在这个例子里, 向量 i 简单地包括了 2~5 的一个序列, 从中我们计算期限矩阵的列指标 (h), 这个指标会用于搜索其他节点。这里的 θ 作为权重来使用。

```
> apply(as.matrix(m[, h]), 2, max) +
+   theta * (apply(as.matrix(m[, h + 1]), 2, max) -
+   apply(as.matrix(m[, h]), 2, max))
```

在这里, 我们找到了期限矩阵每第 h 列中的最大值, 再加上 $h+1$ 列和 h 列的最大结果之差的 θ 加权值, 生成如下的结果。

```
DE0001135101 DE0001141463 DE0001135218 DE0001135317
1.006027      2.380274      5.033425      9.234521
```

现在, 我们 (用 c 函数) 连接第一个值 (最小值) 和之前计算过的最后一个值 (最大值), 这个最大值是之前拷贝上一块代码计算的结果, 这样可以识别所有的节点。

```
> c(floor(min(y[, 1])), apply(as.matrix(m[, h]), 2, max) + theta *
+   (apply(as.matrix(m[, h + 1]), 2, max) - apply(as.matrix(m[, h]), 2,
+   max)), max(m[, ncol(m)]))
DE0001135101 DE0001141463 DE0001135218 DE0001135317
```

```
0.0000      1.006027      2.380274      5.033425      9.234521      31.44657
```

5.5 R 函数应用

在前面的例子里，我们已经使用 `termstrc` 包的一些函数来阐述如何决定理想的节点个数和如何放置这些节点，而本节会使用一些高级 R 函数，以一种更容易的方式来完成这些过程。

```
> x <- estim_cs(govbonds, 'GERMANY')
> x$knotpoints[[1]]
DE0001135101 DE0001141463 DE0001135218 DE0001135317
0.0000      1.006027      2.380274      5.033425      9.234521      31.44657
```

首先，我们使用 `estim_cs` 函数，它基于三次样条估计息票债券的期限结 (Ferstl-Haydn, 2010) 并返回以 `knotpoints` 命名的节点。请注意，`estim_cs` 使用列表工作，这个包里的大多数函数也一样，这就是 `x$knotpoints` 会返回一个列表的原因，这个列表我们仅能检查第一个元素，它等于在前一节中我们动手计算的值。

上述函数会返回其他很多有用的值，缺省结果显示在下面的命令行中。

```
-----
Estimated parameters and robust standard errors:
-----
```

```
[1] "GERMANY:"
```

```
t test of coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
alpha 1	1.9320e-02	1.5230e-02	1.2686	0.2111
alpha 2	-8.4936e-05	3.7926e-03	-0.0224	0.9822
alpha 3	-3.2009e-04	1.1359e-03	-0.2818	0.7794
alpha 4	-3.7101e-04	3.9074e-04	-0.9495	0.3474
alpha 5	7.2921e-04	9.9560e-05	7.3243	3.375e-09 ***
alpha 6	2.0159e-03	1.3019e-04	15.4843	< 2.2e-16 ***
alpha 7	-4.1632e-02	4.5903e-03	-9.0696	1.011e-11 ***

```
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

可以用 `summary` 函数显示出拟合优度, 就像其他的 R 模型一样。

```
-----
Goodness of fit:
-----
```

```

GERMANY
RMSE-Prices      0.198573
AABSE-Prices     0.131036
RMSE-Yields (in %) 0.130108
AABSE-Yields (in %) 0.057223
```

通过对 `x` 对象简单地调用 `plot` 命令, 零息债券的收益率曲线和它的置信区间可以很容易和节点一起显示出来, 如图 5-1 所示。

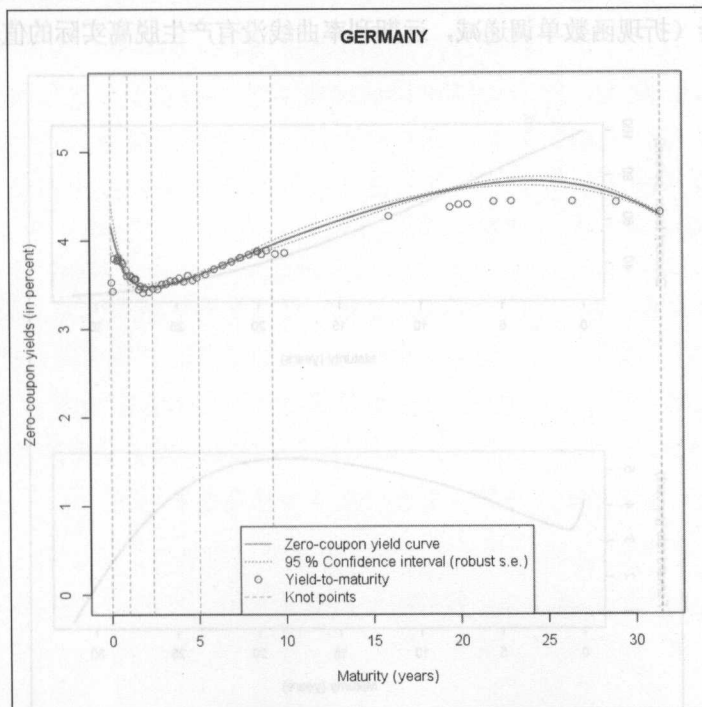


图 5-1 零息债券的收益率曲线

图 5-1 显示了零息债券的收益率曲线估计和数据集中个体证券的到期收益率。两个期限最短的证券是离群点，也许是因为它们在到期之前流动性很小。我们发现，估计的收益率曲线非常接近 10 年期的到期收益率。通常来说，期限更长的零息债券的收益率估计高于息票债券的到期收益率。这也许是拟合不完美的结果，也许是因为事实上这些债券是附息债券而非零息债券。

而且，通过对 `par` 设置 `mfrow`，我们可以在同一张图内画出不同的图像（我们还可以设置 `multiple=TRUE`，这样不必用户输入也可以绘图）。例如，我们可以使用下面的命令，基于 `x` 对象画出折现曲线和远期曲线。

```
> par(mfrow = c(2,1))
> plot(x$discount, multiple = TRUE)
> plot(x$forward, multiple = TRUE)
```

在图 5-2 中，我们看到估计的折现函数和远期利率曲线形态良好，并且没有表现出存在套利机会（折现函数单调递减，远期利率曲线没有产生脱离实际的值和摆动）。

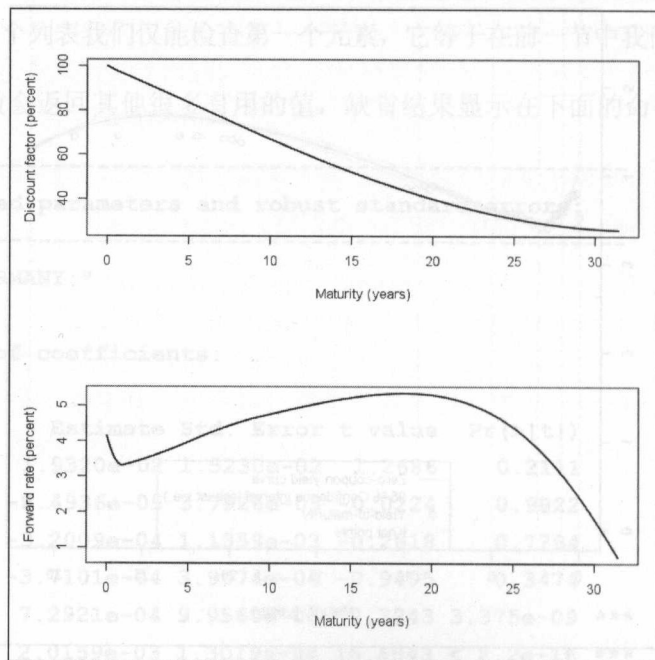


图 5-2 折现曲线和远期曲线

通常来说，期限结构的三次样条估计值很好。但有时候，估计的期限结构并不吸引人（远期利率曲线摆动很剧烈）。这时可以使用非线性样条回归或简约利率曲线模型，但这些内容超过了本章范围。

深入资源，比如 Nelson/Siegel、Diebold/Li、Svensson 以及调整的 Svensson 方法一并提供在 `estim_nss` 函数或 `YieldCurve` 包的帮助中。

5.6 小结

在本章中，我们讨论了通过三次样条回归模型的期限结构估计方法，并展示了如何在 R 中估计利率期限结构。在简要介绍利率和期限结构之后，还讨论了最基本的方法，比如线性回归模型和相关问题，接着详细概述了如何用 R 实现三次样条回归模型，并提到了用于期望任务更复杂情况的已发布的 R 函数和包。

估计的收益率曲线非常接近 10 年期的到期收益率。这也许是不完全正确的，因为收益率估计高于无风险利率的短期收益率。这也许是不完全正确的，因为收益率估计高于无风险利率的短期收益率。这也许是不完全正确的，因为收益率估计高于无风险利率的短期收益率。

第 6 章 衍生品定价

而且，通过对 `par` 设置 `workin` 我们可以在同一窗口中设置 `multiplier`，这样不必用户输入 `multiplier` 以使用下面的命令，基于 `x` 对象画出折现曲线和远期曲线。

本章中，我们讨论了通过三次样条曲线拟合利率期限结构的方法，并展示了

衍生品是一种金融工具，它的价值源于（或者说取决于）称为基础资产的其他产品的价值。衍生品的 3 种基本类型是远期和期货合约、互换，以及期权。本章将关注最后一种类型并讲述基本的期权定价模型，以及一些可以用 R 处理的相关问题。我们始于如何在 R 中使用连续 Black-Scholes 模型和二项 Cox-Ross-Rubinstein 模型，接着继续讨论这些模型之间的联系。进而我们通过计算希腊字母并绘图，说明如何分析期权包含的市场风险的最重要类型。最后，我们会讨论隐含波动率的意义，并通过真实市场数据画出波动率微笑来解释这种现象。

期权区别于期货或互换的最重要特征是，你无法确认这笔交易（买入或者卖出基础资产）是否会发生。这种特征使期权定价更加复杂，并要求所有的模型对基础产品的未来价格变动做出假设。这里涉及的两种模型在假设上有所区别：Black-Scholes 模型采用连续过程，而 Cox-Ross-Rubinstein 模型采用离散的随机过程。但是，其余假设非常相似，并且结果也很接近（甚至基本相同）。

6.1 Black-Scholes 模型

Black-Scholes 模型（Black 和 Sholes, 1973，也见 Merton, 1973）的假设如下。

- 基础资产的价格 (S) 服从几何布朗运动: $dS = \mu S dt + \sigma S dW$, 其中, μ (漂移率) 和 σ (波动率) 是常数参数, W 是一个标准的维纳过程。
- 市场无套利。
- 基础产品是不支付红利的股票。
- 可以以任意数量 (甚至是分数) 买入或者 (做空) 卖出基础资产。
- 没有交易成本。
- 短期利率 (r) 已知, 并且不随着时间变动。

模型的主要结果是, 在这些假设之下, 欧式看涨期权的价格 (c) 有闭式的公式。

$$c = SN(d_1) - Xe^{-r(T-t)}N(d_2)$$

$$d_1 = \frac{\ln \frac{S}{X} + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

在这里, X 是执行价格, $T-t$ 是期权到期的时间, N 表示标准正态分布的累积分布函数。这个方程给出的期权价格就是通常所说的 Black-Scholes 公式。从看跌-看涨平价关系中容易看出, 参数相同的欧式看跌期权的价格 (p) 由下式给出。

$$p = Xe^{-r(T-t)}N(-d_2) - SN(-d_1)$$

现在, 在 2013 年 6 月, 考虑基于谷歌股票的一个看涨期权和一个看跌期权, 到期日为 2013 年 9 月 (就是说, 距离到期有 3 个月的时间)。我们假设基础股票的当前价格是 900 美元, 执行价格是 950 美元, 谷歌的波动率是 22%, 无风险利率是 2%。我们使用 fOptions 包的 GBSOption 函数来计算看涨期权值。除了已经讨论过的参数, 我们还需要考虑持有成本 (b)。在原始的 Black-Scholes 模型中, 它等于无风险利率 (基础资产不支付红利)。

```
> library(fOptions)
> GBSOption(TypeFlag = "c", S = 900, X = 950, Time = 1/4, r = 0.02,
+   sigma = 0.22, b = 0.02)
Title:
  Black Scholes Option Valuation
Call:
  GBSOption(TypeFlag = "c", S = 900, X = 950, Time = 1/4, r = 0.02,
    b = 0.02, sigma = 0.22)
```

Parameters:

Value:

TypeFlag c

S 900

X 950

Time 0.25

r 0.02

b 0.02

sigma 0.22

Option Price:

21.79275

Description:

Tue Jun 25 12:54:41 2013

长长的输出返回了传递参数，结果在 Option Price 标签之下。把 TypeFlag 设置为 p 会计算看跌期权的价格，而现在我们只关心结果（可在 price 项中发现，更多细节请参见 str 对象），不关心原文的输出。

```
> GBSOption(TypeFlag = "p", S = 900, X = 950, Time = 1/4, r = 0.02, sigma
= 0.22, b = 0.02)@price
[1] 67.05461
```

正如在前一章，我们也可以选择使用 GUIDE 包提供的界面更加用户友好的计算器，计算上面的值。运行 blackscholes() 函数会触发一个模型窗口，我们可

以输入相同的参数。请注意，这个函数使用了红利率代替了持有成本，它在本例中为 0。

6.2 Cox-Ross-Rubinstein 模型

Cox-Ross-Rubinstein (CRR) 模型 (Cox, Ross 和 Rubinstein, 1979) 假定基础资产的价格服从离散的二项过程。在每个时期，价格可以向上也可以向下，并根据图 6-1 所示的二项树变动，其中 u 和 d 表示固定乘子，分别测量价格向上和向下的变动。CCR 模型的重要特征是 $u=1/d$ 并且树是重组的。就是说，如果价格先上升再下降（反之结果一样），价格在两个时期之后与初始时相同，这显示在图 6-1 中。

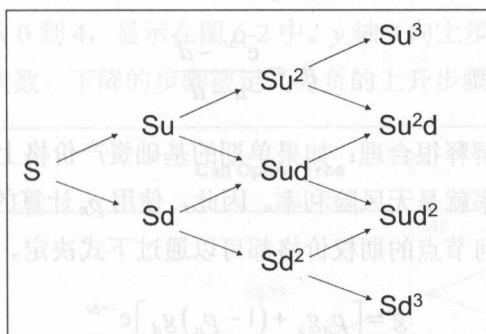


图 6-1 CCR 模型的二项树

为了建立一个二项树，我们首先需要决定建模的步骤个数 (n)，即，距离期权到期的时间需要被划分为多少步骤。或者，我们也可以决定树的一个时间步骤的长度 Δt (用年表示)。

$$\Delta t = \frac{T-t}{n}$$

如果我们已知基础资产的波动率 (σ)，参数 u 和 d 可以根据下式来决定。

$$u = e^{\sigma\sqrt{\Delta t}}$$

即:

$$d = e^{-\sigma\sqrt{\Delta t}}$$

当使用二项模型对期权定价时, 我们需要确定基础资产直到到期时的树。于是, 有了到期时所有可能的价格, 我们可以计算对应的可能期权的价值, 由下列公式简单地给出。

$$c_T = \max(0, S_T - X)$$

$$p_T = \max(0, X - S_T)$$

为了使用二项模型决定期权的价格, 我们需要在每个节点计算接下来两个可能的期权价值的期望值, 并把它们折现。问题在于折现使用的预期收益率不平凡。解决的技巧是使用一种假设的概率来计算预期收益, 这样我们可以使用无风险利率进行折现。这种概率称为风险中性概率 (p_n), 通过下面的公式给出。

$$p_n = \frac{e^{r\Delta t} - d}{u - d}$$

风险中性概率的解释很合理: 如果单期的基础资产价格上涨的概率是 p_n , 那么基础资产的预期收益率就是无风险利率。因此, 使用 p_n 计算的预期值可以通过 r 折现, 并且二项树的任何节点的期权价格都可以通过下式决定。

$$g = [p_n g_u + (1 - p_n) g_d] e^{-r\Delta t}$$

在上述公式中, g 是在给定的节点上的一般期权价格 (可以是看涨期权也可以是看跌期权), g_u 和 g_d 表示这个衍生品在一个时期之后的两个可能节点上的价值。

为了在 R 中演示 CRR 模型, 我们将使用与 Black-Scholes 公式案例相同的参数, 因此, $S=900$, $X=950$, $\sigma=22\%$, $r=2\%$, $b=2\%$, $T-t=0.25$ 。我们还需要设置 n , 二项树的时间步长的个数。为了说明, 我们将使用一个三期的模型。

```
[1] 67.05461
> CRRBinomialTreeOption(TypeFlag = "ce", S = 900, X = 950,
+ Time = 1/4, r = 0.02, b = 0.02, sigma = 0.22, n = 3)@price
[1] 20.33618
> CRRBinomialTreeOption(TypeFlag = "pe", S = 900, X = 950,
```



```
+ Time = 1/4, r = 0.02, b = 0.02, sigma = 0.22, n = 3)@price
[1] 65.59803
```

值得观察的是, 从二项模型获得的期权价格非常接近 (但并非精确相同) 前面计算的 Black-Scholes 价格。除了最终结果, 也就是期权的当前价格, 我们还对整个期权树有兴趣。

```
> CRRTree <- BinomialTreeOption(TypeFlag = "ce", S = 900, X = 950,
+ Time = 1/4, r = 0.02, b = 0.02, sigma = 0.22, n = 3)
> BinomialTreePlot(CRRTree, dy = 1, xlab = "Time steps",
+ ylab = "Number of up steps", xlim = c(0,4))
> title(main = "Call Option Tree")
```

在这里, 我们先通过参数给定的 BinomialTreeOption 函数计算一个矩阵, 并把结果保存在 CRRTree 中, 再传递给绘图函数, 图的 x 轴和 y 轴的名称都已指定, 并且 x 轴限制为从 0 到 4, 显示在图 6-2 中。 y 轴 (向上步长的个数) 表示基础资产价格总共上升的次数。下降的步骤被定义为负的上升步骤。

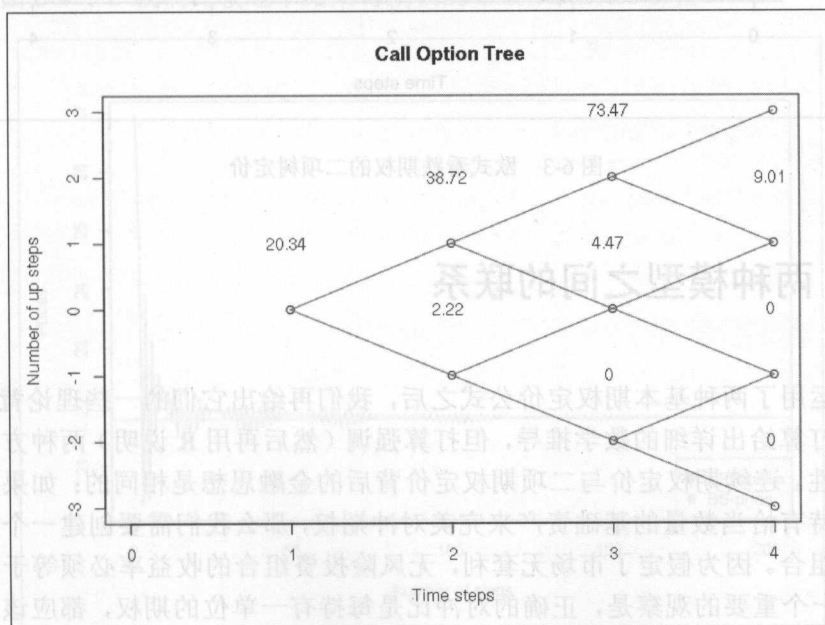


图 6-2 欧式看涨期权的二项树定价

在前面的代码中, 改变 `TypeFlag` 为 `pe`, 图 6-3 可以相似地显示出欧式看跌期权。

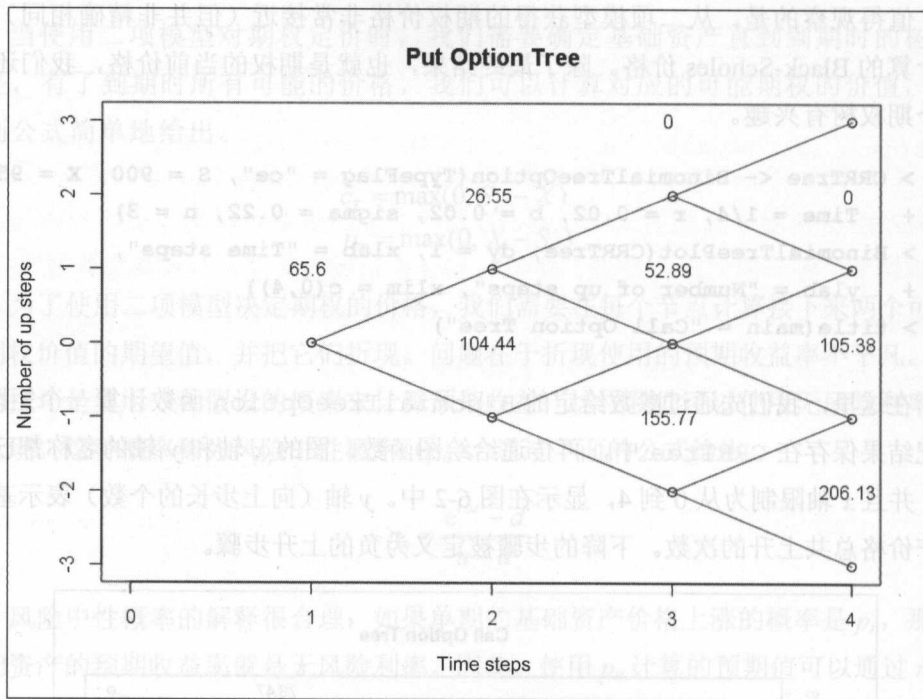


图 6-3 欧式看跌期权的二项树定价

6.3 两种模型之间的联系

在运用了两种基本期权定价公式之后, 我们再给出它们的一些理论背景。我们并不打算给出详细的数学推导, 但打算强调 (然后再用 `R` 说明) 两种方法之间的相似性。连续期权定价与二项期权定价背后的金融思想是相同的: 如果我们希望通过持有恰当数量的基础资产来完美对冲期权, 那么我们需要创建一个无风险的投资组合。因为假定了市场无套利, 无风险投资组合的收益率必须等于无风险利率。一个重要的观察是, 正确的对冲比是每持有一单位的期权, 都应该相应地持有 $\frac{\partial g}{\partial S}$ 单位的基础资产。因此, 这个比例是期权价格相对于基础资产价格的偏

导数（或者在二项模型中的离散对应形式）。这个偏导数叫作期权的 **delta**。两个模型之间还有另一种有趣的联系，是 **delta** 对冲策略和相关的无套利原理产生了相同的定价原则：衍生品的价值是它的未来价值经过无风险利率折现的风险中性预期值。这个原则在二项树上容易处理，我们可以一个节点一个节点地计算折现的预期值。而且，连续模型的逻辑也相同，虽然它的预期值的数学计算更复杂。这就是为什么我们仅仅给出这种原则最终结果的原因，这种结果是 **Black-Scholes** 公式。

现在，我们知道了两个模型有同样的定价原则和思想（**delta** 对冲和风险中性估值），但我们依然观察到它们的数值结果并不相等。原因是用来描述基础资产价格运动的假设随机过程不同。尽管如此，它们仍然非常相似。如果我们决定了来自波动率参数的 u 和 d 的值，正如我们在 **Cox-Ross-Rubinstein** 这一节中的所为，那么二项过程近似于几何布朗运动。因此，如果我们增加时间步长的个数（或者等价的，减少步长的长度），二项模型的期权价格会收敛到 **Black-Scholes** 模型的价格。

为了说明这种联系，我们用增加的时间步长的个数来计算二项模型中的期权价格。在图 6-4 中，我们比较这个结果与期权的 **Black-Scholes** 价格。

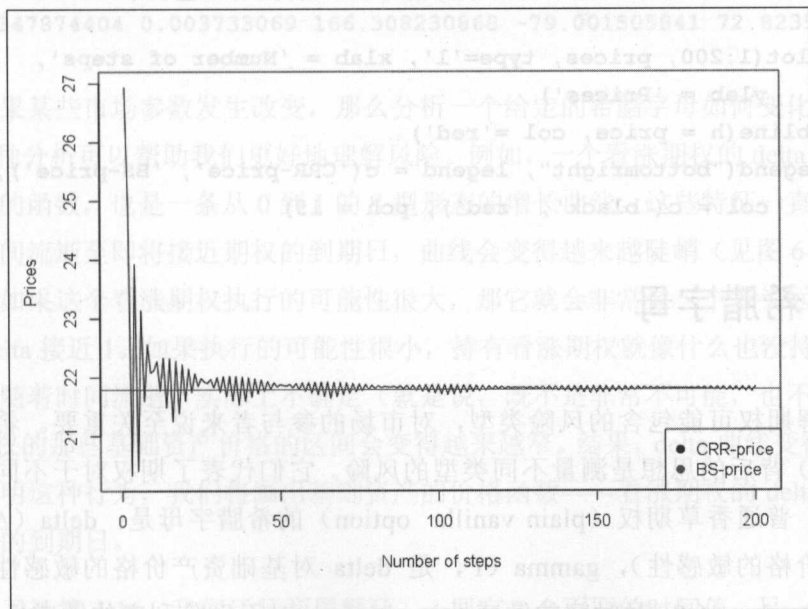


图 6-4 两种定价公式结果的比较

对 N 从 1 到 200 运行循环来计算参数固定的 CRRBinomialTreeOption, 生成了图 6-4。

```
> prices <- sapply(1:200, function(n) {
+   CRRBinomialTreeOption(TypeFlag = "ce", S = 900, X = 950,
+     Time = 1/4, r = 0.02, b = 0.02, sigma = 0.22, n = n)@price
+ })
```

现在, prices 变量包括了 200 个计算出来的值。

```
> str(prices)
num [1:200] 26.9 24.9 20.3 23.9 20.4...
```

我们也来计算一般 Black-Scholes 的期权。

```
> price <- GBSOption(TypeFlag = "c", S = 900, X = 950, Time = 1/4, r =
0.02, sigma = 0.22, b = 0.02)@price
```

接着在一张图中联合显示这两个价格, GBS 期权用红色表示。

```
> plot(1:200, prices, type='l', xlab = 'Number of steps',
+   ylab = 'Prices')
> abline(h = price, col = 'red')
> legend("bottomright", legend = c('CRR-price', 'BS-price'),
+   col = c('black', 'red'), pch = 19)
```

6.3 两种模型之间的联系

6.4 希腊字母

理解期权可能包含的风险类型, 对市场的参与者来说至关重要。希腊字母 (Greeks) 背后的思想是测量不同类型的风险。它们代表了期权对于不同因素的敏感性。普通香草期权 (plain vanilla option) 的希腊字母是: delta (Δ , 对基础资产价格的敏感性), gamma (Γ , 是 delta 对基础资产价格的敏感性, delta 的 delta), theta (θ , 对时间的敏感性), rho (ρ , 对无风险利率的敏感性), vega (V , 对波动率的敏感性)。用数学语言来说, 所有的希腊字母都是衍生品价格

的偏导数。

$$\Delta = \frac{\partial g}{\partial S}$$

$$\Gamma = \frac{\partial^2 g}{\partial S^2}$$

$$\theta = \frac{\partial g}{\partial t}$$

$$\rho = \frac{\partial g}{\partial r}$$

$$V = \frac{\partial g}{\partial \sigma}$$

使用 GBSGreeks 函数可以容易计算每个期权的希腊字母。

```
> sapply(c('delta', 'gamma', 'vega', 'theta', 'rho'), function(greek)
+   GBSGreeks(Selection = greek, TypeFlag = "c", S = 900, X = 950,
+     Time = 1/4, r = 0.02, b = 0.02, sigma = 0.22)
+ )
```

delta	gamma	vega	theta	rho
0.347874404	0.003733069	166.308230868	-79.001505841	72.82355323

如果某些市场参数发生改变，那么分析一个给定的希腊字母如何变化通常很有用。这种分析可以帮助我们更好地理解风险。例如，一个看涨期权的 delta 是基础资产价格的函数，也是一条从 0 到 1 的 S 型形态的增长曲线。这些特征一直有效，但如果时间流逝至即将接近期权的到期日，曲线会变得越来越陡峭（见图 6-5）。原因如下：如果这个看涨期权执行的可能性很大，那它就会非常像一个多头远期合约，因此 delta 接近 1。如果执行的可能性很小，持有看涨期权就像什么也没持有，delta 是 0。随着时间流逝，实际上不确定（就是说，既不是非常不可能，也不是非常可能）行权的那些基础资产价格的区间会变得越来越窄。结果，delta 曲线变得更陡峭。为了阐明这种行为，我们将画出基础资产的价格函数——看涨期权的 delta，使用 3 个不同的到期日。

为了计算 delta，我们运行两层循环：一层有 3 个不同的时间值，另一层对 s 从 500 运行到 1500。


```

> deltas <- sapply(c(1/4, 1/20, 1/50), function(t)
+               sapply(500:1500, function(S)
+               GBSGreeks(Selection = 'delta', TypeFlag = "c",
+               S = S, X = 950, Time = t, r = 0.02, b = 0.02, sigma = 0.22)))

```

得到的 deltas 结果包括有 1001 行（对应 S 的值）和 3 列（对应指定的时间），我们把它都画在一张图中（见图 6-5）。

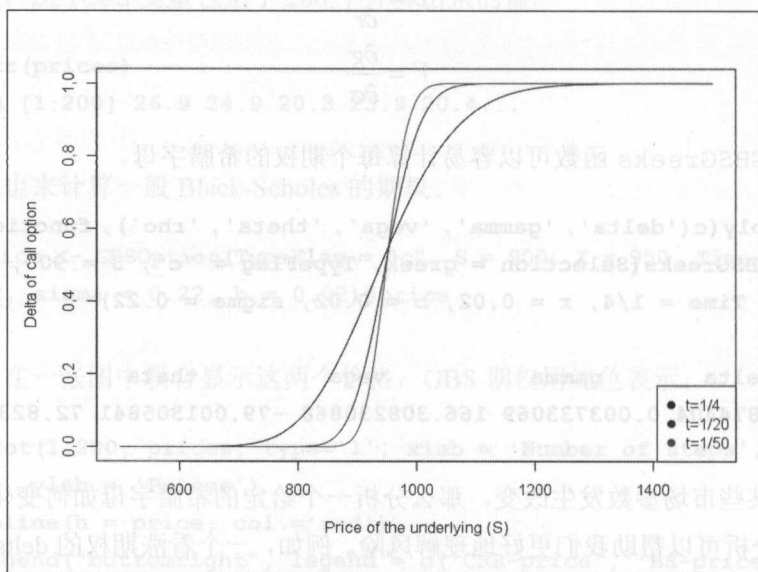


图 6-5 看涨期权的 delta

```

> plot(500:1500, deltas[, 1], ylab = 'Delta of call option',
+      xlab = "Price of the underlying (S)", type = 'l')
> lines(500:1500, deltas[, 2], col='blue')
> lines(500:1500, deltas[, 3], col='red')
> legend("bottomright", legend = c('t=1/4', 't=1/20', 't=1/50'),
+      col = c('black', 'blue', 'red'), pch = 19)

```

图 6-5 显示了 3 个不同到期时间值的看涨期权的 delta。

确定或者画出复杂期权策略的希腊字母非常相似。例如，计算一个跨式头寸（相

同参数的一个看涨期权和一个看跌期权的投资组合)意味着分别对看涨期权和看跌期权仅仅计算 delta, 接着再求和。我们会把跨式期权的 delta 绘作基础资产价格的函数。可以观察到这个图像的形狀类似于之前看涨期权的 delta, 但现在的 S 曲线的范围是从-1 到 1。

```
> straddles <- sapply(c('c', 'p'), function(type)
+               sapply(500:1500, function(S)
+               GBSGreeks(Selection = 'delta', TypeFlag = type, S = S,
+               X = 950, Time = 1/4, r = 0.02, b = 0.02, sigma = 0.22)))
```

所以我们调用一个嵌套的循环, 看涨期权和看跌期权都让 S 从 500 运行到 1500, 同时保持其他参数不变, 并将得到的 delta 保存在一个矩阵中。通过下面的命令, 结果提供了这些行的和 (看跌和看涨期权)。

```
> plot(500:1500, rowSums(straddles), type='l',
+      xlab='Price of the underlying (S)', ylab = 'Delta of straddle')
```

结果图刻画了跨式头寸的 delta, 它是基础资产的价格函数, 展示在图 6-6 中。

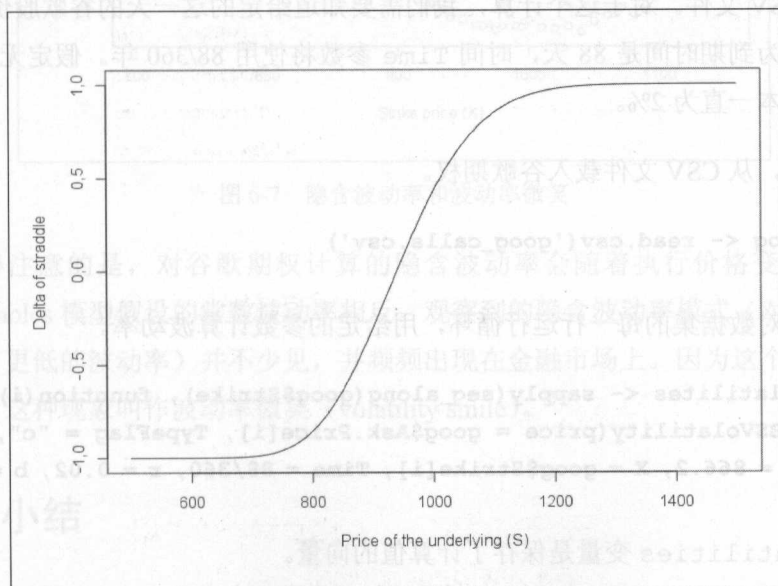


图 6-6 一个跨式组合的 delta

6.5 隐含波动率

Black-Scholes 模型因为存在一些缺点，常常受到批评。一个重要的问题是，这个模型假设基础资产的波动率是常数，这在现实中并不成立。而且，因为波动率 *volatility* 无法直接观察，因而是模型中最难校准的参数。由于这种困难，Black-Scholes 公式通常使用一种间接方式来估计 *volatility* 参数。我们观察一个期权的市场价格，由于其他的所有参数都可以观察到，我们寻找使 Black-Scholes 价格等于市场价格观测值的 σ 。这个 σ 参数叫作期权的隐含波动率。正如 Riccardo Rebonato 的名言，隐含波动率是“一个放到错误的公式中，得到正确价格的错误的数” (Rebonato, 1999, p.78)。

我们借助于一些谷歌期权，将说明隐含波动率的计算。期权是到期日为 2013 年 9 月 21 日的看涨期权，执行价格在 750 美元到 1150 美元之间(76 个不同的期权)。我们从 finance.google.com 收集这些期权在 2013 年 6 月 25 日的卖价，再把它们存储到一个 CSV 文件。对于这个计算，我们需要知道给定的这一天的谷歌股价为 866.2 美元。因为到期时间是 88 天，时间 *Time* 参数将使用 88/360 年。假定无风险利率和持有成本一直为 2%。

首先，从 CSV 文件载入谷歌期权。

```
> goog <- read.csv('goog_calls.csv')
```

接着对数据集的每一行运行循环，用给定的参数计算波动率。

```
> volatilities <- sapply(seq_along(goog$Strike), function(i)
+   GBSVolatility(price = goog$Ask.Price[i], TypeFlag = "c",
+   S = 866.2, X = goog$Strike[i], Time = 88/360, r = 0.02, b = 0.02))
```

volatilities 变量是保存了计算值的向量。

```
> str(volatilities)
```

```
num [1:76] 0.258 0.253 0.269 0.267 0.257...
```

可以针对执行价格进行显示。

```
> plot(x = goog$Strike, volatilites, type = 'p',
+      ylab = 'Implied volatility', xlab = 'Strike price (X)')
```

因此，图 6-7 展示了不同的执行价格对应的隐含波动率。

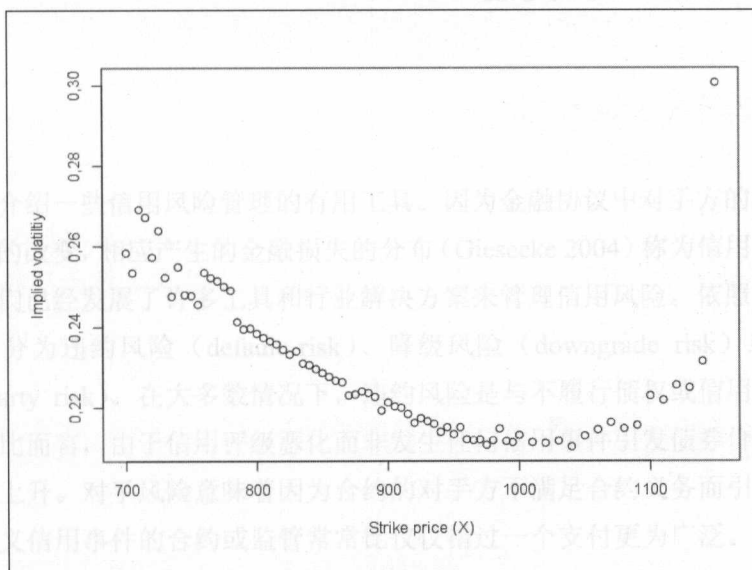


图 6-7 隐含波动率和波动率微笑

值得注意的是，对谷歌期权计算的隐含波动率会随着执行价格变动。这与 Black-Scholes 模型假设的常数波动率相反。观察到的隐含波动率模式（对中等执行价格有更低的波动率）并不少见，并频频出现在金融市场上。因为这个曲线的特定形式，这种现象叫作波动率微笑（volatility smile）。

6.6 小结

在本章中，我们使用 R 并采用 Black-Scholes 和 Cox-Ross-Rubinstein 两种模型

对普通香草期权做定价。并且，我们检查了这些期权的希腊字母和隐含波动率。这些主题的更多金融背景细节，参见（Hull，2011）。除了需要了解 fOptions 包的一些工具，为了进行模拟，我们还通过编程的方式创建了一些循环和自定义的函数。下一章将专心应用各种模型来管理信用风险，比如采用蒙特卡洛模拟方法和信用得分方法来选择最优的信用投资组合。

Black-Scholes 公式通常使用一种间接方式来估计 volatility 参数。我们观察一个期权的市场价格。由于其他的所有参数都可以观察到，我们寻找使 Black-Scholes 价格等于市场价格观测值的 σ 。这个 σ 参数叫作期权的隐含波动率。正如 Riccardo Rebonato 的名言，隐含波动率是“一个放到错误的公式中，得到正确价格的错误的数”（Rebonato，1999，p.78）。

我们借助于一些谷歌期权，将说明隐含波动率的计算。期权是到期日为 2013 年 9 月 21 日的看涨期权，执行价格在 750 美元到 1150 美元之间（76 个不同的期权）。我们从 finance.google.com 收集这些期权在 2013 年 6 月 25 日的报价，再把它们存储到一个 CSV 文件。对于这个计算，我们需要知道给定的这一天的谷歌股价为 866.2 美元。因为到期时间是 88 天，我们使用了 180 天数据率使用 8.8355%。假定无风险利率和持仓成本一直为 2%。



图 6.7 从 CSV 文件读入谷歌期权
隐含波动率与行权价格关系图

```
> goog <- read.csv('goog_calls.csv')
# 隐含波动率计算
# 注意：这里隐含波动率是随着行权价格变化的，这是因为 Black-Scholes 模型假设波动率是常数，而实际上波动率是随价格变化的。因此，我们使用隐含波动率来计算期权价格。
# 计算隐含波动率
volatility <- as.list(mapply(function(x, y) {
  # 计算隐含波动率
  # 使用 bisection 方法
  # 初始猜测
  c <- 0.1
  # 迭代
  for (i in 1:100) {
    # 计算期权价格
    price <- BSPrice(x, y, c)
    # 比较期权价格与市场价格
    if (abs(price - y) < 0.01) {
      return(c)
    }
    # 更新猜测
    c <- (c + y/price) / 2
  }
  return(c)
}, strike, price))
# 输出
# 隐含波动率
volatility
```

在本章中，我们使用 R 来计算 Black-Scholes 和 Cox-Ross-Rubinstein 模型。

第 7 章

信用风险管理

本章介绍一些信用风险管理的有用工具。因为金融协议中对手方的信用质量发生未预期的改变，相应产生的金融损失的分布 (Giesecke 2004) 称为信用风险 (credit risk)。人们已经发展了许多工具和行业解决方案来管理信用风险。依照文献，信用风险可以分为违约风险 (default risk)、降级风险 (downgrade risk) 或对手风险 (counterparty risk)。在大多数情况下，违约风险是与不履行债权或信用直接相关的风险。相比而言，由于信用评级恶化而非发生任何信用事件引发债券价格下降时，降级风险上升。对手风险意味着因为合约的对手方不满足合约义务而引发的风险。但是，定义信用事件的合约或监管常常比仅仅错过一个支付更为广泛。对可能违约的最终估计建模是所有 3 种情况的基本要求。

信用风险管理在金融机构中通过多种方式实施。通常的相关任务如下。

- 信用组合选择 (例如，商业银行关于贷款或信用评分的决策)。
- 度量并预测违约或降级的概率 (比如，使用 CreditMetrics 的信用评级迁移矩阵)。
- 对违约或降级所引起的金融损失分布建模 (对单个实体进行结构化的和简约化的定价和建模，或者对投资组合进行相依性的结构化建模)。
- 减轻或消除信用风险 (采用对冲、分散化、预防，或者保险，我们不会在本

书中进行研究)。

在本章中,我们对上面列出的一些问题会给出一些 R 应用案例。首先,我们介绍信用损失建模的基本概念,即结构化和简约化的方法,及其在 R 中的应用。接着,我们提供一种叫作 copula 的相关随机变量实践方法,它对结构化信用衍生品定价是一种有用的技术。我们还会说明如何使用 R 管理信用迁移矩阵,最后,我们还使用分析工具深入探究信用得分,比如 logit 和 probit 回归以及受试者操作特征曲线(ROC)分析。

7.1 信用违约模型

本章第一部分的目的是,显示使用 R 来对标准的信用风险模型定价和实施蒙特卡洛模型的方法。下面各小节概述了损失分布和单一债务产品的生成和定价。

7.1.1 结构模型

我们从著名的莫顿期权模型(Merton 1974)开始,它是结构方法的基本模型。莫顿把风险债务作为公司价值的或有要求权(contingent claim)进行评估。我们假设公司价值 V 服从几何布朗运动。

$$dV_t = \mu V_t + \sigma V_t dW_t$$

在上面的公式中, μ 是漂移参数, σ 是波动率参数, dW 是维纳过程的微分,初始资产值是 V_0 。模型假设一条平坦的收益率曲线,把 r 作为常数利率。如果资产 V 的价值低于期限债务(T)的负债(K),我们把这样的情况视为违约状态。我们把到期时的公司价值 V_T 表示为一个积分。

$$V_T = V_0 \exp\left(\int_0^T d\ln V_t\right)$$

其中我们使用 Ito 引理来表示 $d\ln V_t$ ，导出公司价值的对数微分。

$$d\ln V_t = \left(\mu - \frac{\sigma^2}{2} \right) dt + \sigma dW_t$$

除了为捕捉 $\Delta W \sim \sqrt{\Delta t}N(0,1)$ 生成正态分布的随机变量，我们还通过下面的方式使用离散方法计算到期的公司价值 V_T 。

$$V_T = V_0 \exp \left(\sum_{i=1}^M \left(\mu - \frac{\sigma^2}{2} \right) \Delta t + \sigma \Delta W_i \right)$$

其中的 Δt 表示一个时期经过的时间长度。基于这种逻辑，我们用 R 来模拟公司价值。首先，按照下面的方式设置模拟参数，即初始的资产价值、漂移项以及波动率参数。

```
> V0 <- 100; nu <- 0.1; sigma <- 0.2
```

接下来，声明 Δt 的长度和时间周期的终点 (Time)。

```
> dt <- 1 / 252; Time <- 1
```

还要计算时间周期的个数。

```
> M <- Time / dt
```

最后，决定生成轨线的个数。

```
> n <- 10000
```

为了你将来有可能复制生成的伪随机变量，每次调用生成函数前需要设置随机数种子。

```
> set.seed(117)
```

以及，为了产生过程 V 的对数增量 ($\Delta \ln V$)，需要从正态分布中使用指定的均值和标准差生成 $n \times M$ 个随机数。

```
> val <- rnorm(n*M,
+   mean = (nu - sigma^2 / 2) * dt,
+   sd = sigma * dt^0.5)
```

把这些数存储在一个 M 行 n 列的矩阵中。

```
> dlnV <- matrix(val, M, n)
```

为了及时得到公司价值的变化 (V)，需要汇总增量的对数，正如上面的计算，取这个和的对数并乘以初始值。计算 n 条轨线的代码如下。

```
> V <- V0 * exp(apply(dlnV, 2, cumsum))
```

这里使用的 `cumsum` 函数常常与 `apply` 命令结合在一起。用 `matplot` 函数画前 5 条轨线，如下：

```
> matplot(x = seq(0 + dt, Time, dt), y = V[, 1:5], type = 's', lty = 1,
+   xlab = 'Time',
+   ylab = 'Firm value trajectories',
+   main = 'Trajectories of firm values in the Merton model')
```

我们选择用实 (`lty=1`) 阶梯线 (`type='s'`) 画出 V 的前 5 列 (`1:5`)，如图 7-1 所示。

注意，在风险中性或者鞅 P 测度之下，风险债务的价格是其在到期时支付的折现期望值，其中漂移项是无风险利率 r ，如下面的公式所示。

$$D = E^P \left[e^{-rT} \min(V_T, K) \right]$$

如果我们设置了无风险利率 r 和债务的面值 (K)。

```
> r <- 0.05; K <- 80
```

我们可以说明，正如之前的公式所示，我们使用蒙特卡洛定价计算债务面值和 V 中第 M 列的平行最小值，再取均值，我们在之前曾经计算过这个 V ，就像下

面这样。

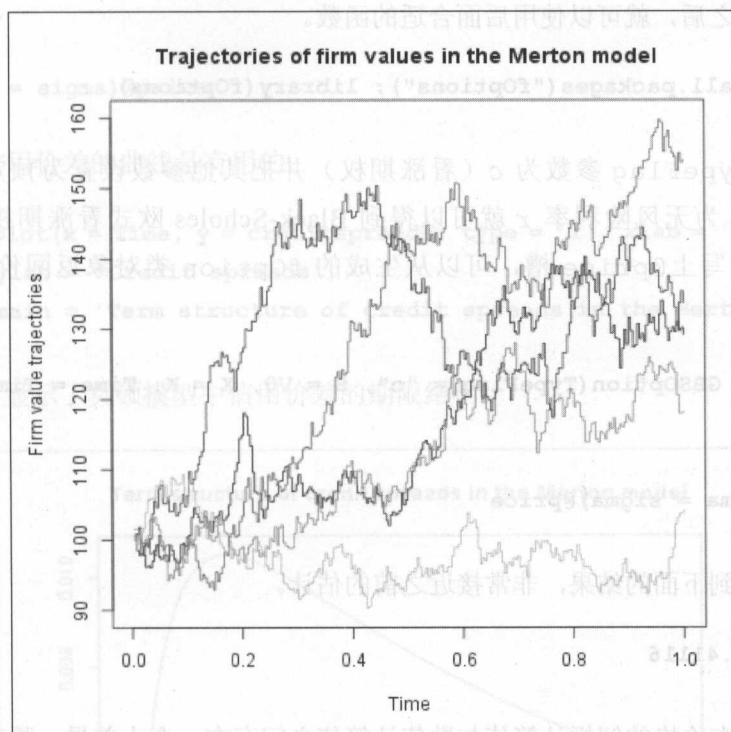


图 7-1 莫顿模型中公司值的轨线

```
> D <- exp(-r * Time) * mean((pmin(V[M, ], K)))
```

对标准参数以及生成的伪随机固定变量 D ，面值为 80 美元的风险债务值为。

```
> D
```

```
[1] 75.73553
```

根据欧式看涨期权的 Black-Scholes 定价公式，时刻 $t=0$ 的风险债务的价值可以表示为公司价值 V 减去股权价值 (E)， E 是 V 上的一个欧式看涨期权。注意 c^{BS} 的定价公式，我们会得到：

$$D = V - c^{BS}(V, K, r, \sigma, T)$$

可以使用 fOptions 包的 GBSOption 函数来计算债务的价值。在安装并调用了下面的包之后, 就可以使用后面合适的函数。

```
> install.packages("fOptions"); library(fOptions)
```

设置 TypeFlag 参数为 c (看涨期权) 并把其他参数设置为预定义的值。选择参数 b 为无风险利率 r 就可以得到 Black-Scholes 欧式看涨期权, 并且在命令的最后写上 @price 槽, 可以从生成的 fOption 类对象返回价格 price 的值。

```
> V0 - GBSOption(TypeFlag = "c", S = V0, X = K, Time = Time, r = r,
+
+ b =
+ r, sigma = sigma)@price
```

我们得到下面的结果, 非常接近之前的估计。

```
[1] 75.41116
```

但是, 在价格的解析计算值与数值计算值之间存在一个小差异。随着轨线数目的增加和 Δt 的减小, 蒙特卡洛模拟的计算价格会收敛到理论价格。在 $t=0$ 时刻, 按照下面的公式, 我们基于风险债务计算信用价差的期限结构 (由 $s(T)$ 表示), 其中信用价差依赖于债务的到期日 (T)。

$$s(T) = \frac{1}{T} \ln \left(\frac{K}{D} \right) - r$$

对不同的期限 (从 0.1 到 10 年, 间隔 0.1 年), 在一条驼峰形状的曲线上画出信用价差。时间格点如下定义。

```
> Time <- seq(0.1, 10, 0.1)
```

接着, 为了计算信用价差, 在格点上的每个点重新计算债务价值。

```
> D <- V0 - GBSOption(TypeFlag = "c", S = V0, X = K, Time = Time, r = r,
```

```

+
b = r,

sigma = sigma)@price

```

画出信用价差的曲线是有用的。

```

> matplot(x = Time, y = creditspreads, type = 'l', xlab = 'Maturity',
+         ylab = 'Credit spreads',
+         main = 'Term structure of credit spreads in the Merton model')

```

图 7-2 显示了莫顿模型中信用价差的期限结构。

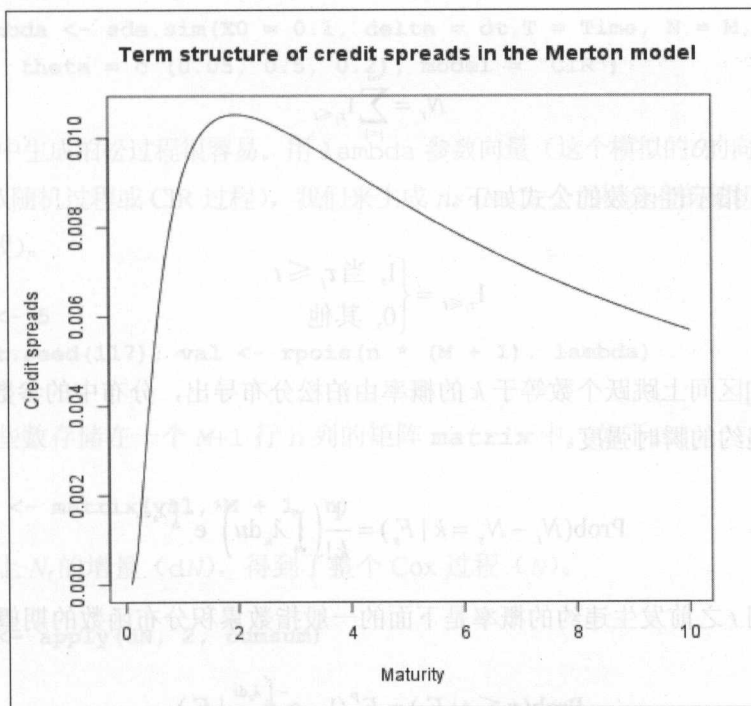


图 7-2 莫顿模型中信用价差的期限结构

莫顿模型有多种扩展，比如，穆迪（Moody）的 KMV 应用，或者布莱克（Black）和考克斯（Cox）的首次命中时间方法。

可以使用 ROptions 包的 GSOOption 函数来计算债务的价值。在安装并调用

7.1.2 强度模型

```
> install.packages("ROptions"); library(ROptions); sample@sigma = sigma
```

信用违约建模的第二个方法既普通又主流，是基于强度（简约形式）的建模。在该框架内，我们假定还没有任何早期的预测信号出现违约就发生了。违约强度的动态取决于违约的预测概率。我们还会在本章后面给出一个强度为常数的例子。

强度模型背后的中心思想是，在给定时间内独立违约的个数服从泊松分布。令 $\tau_1, \tau_2, \dots, \tau_i, \dots, \tau_n$ 表示随机违约时间。因此，令 N_t 表示直到时间 t 违约的个数。

$$N_t = \sum_{i=1}^{\infty} 1_{\tau_i \leq t}$$

其中，指示性函数的公式如下。

$$1_{\tau_i \leq t} = \begin{cases} 1, & \text{当 } \tau_i \leq t \\ 0, & \text{其他} \end{cases}$$

在 $[s, t]$ 区间上跳跃个数等于 k 的概率由泊松分布导出，分布中的参数 λ_u 是在时间 u 发生违约的瞬时强度。

$$\text{Prob}(N_t - N_s = k | F_s) = \frac{1}{k!} \left(\int_s^t \lambda_u du \right)^k e^{-\int_s^t \lambda_u du}$$

在时间 t 之前发生违约的概率是下面的一般指数累积分布函数的期望值。

$$\text{Prob}(\tau \leq t | F_0) = E^P(1 - e^{-\int_0^t \lambda_s ds} | F_0)$$

但是，尽管基本模型使用了常数风险比 λ ，行业模型仍然采用了更复杂的结构。比如，在双随机泊松模型〔或者考克斯（Cox）模型〕中，风险比服从下面等式描述的 Cox-Ingersoll-Ross 过程。

$$d\lambda = (\theta_1 - \theta_2 \lambda_t)dt + \theta_3 \sqrt{\lambda_t} dW$$

sde 包提供了 Cox-Ingersoll-Ross (CIR) 过程的仿真。

```
> library(sde)
```

重新定义时间度量、到期日、长度以及时间周期个数。

```
> Time <- 1; dt <- 1/252; M <- Time / dt
```

然后创建 CIR 过程, 声明了初始值 x_0 , 漂移率参数 θ_1 和 θ_2 (θ_1/θ_2 是长期值, θ_2 是调整的速度), 以及波动率参数 θ_3 。

```
> lambda <- sde.sim(X0 = 0.1, delta = dt, T = Time, N = M,
+   theta = c(0.05, 0.5, 0.2), model = "CIR")
```

在 R 中生成泊松过程很容易。用 `lambda` 参数向量 (这个模拟的 θ 的向量证明了过程是一个双随机过程或 CIR 过程), 我们来生成 $n \times (M+1)$ 个泊松分布的随机变量 (n 是轨线的个数)。

```
> n <- 5
```

```
> set.seed(117); val <- rpois(n * (M + 1), lambda)
```

把这些数存储在一个 $M+1$ 行 n 列的矩阵 `matrix` 中, 如下。

```
> dN <- matrix(val, M + 1, n)
```

再加上 N_t 的增量 (`dN`), 得到了整个 Cox 过程 (N)。

```
> N <- apply(dN, 2, cumsum)
```

按照本章前面的方式把它画出来, 得到的图形见图 7-3。

```
> matplot(x = seq(0, Time, dt), y = N[, 1:5], type = 's', xlab = 'Time',
+   ylab = "'Number of defaults' process trajectories",
+   main = 'Trajectories of Cox processes')
```

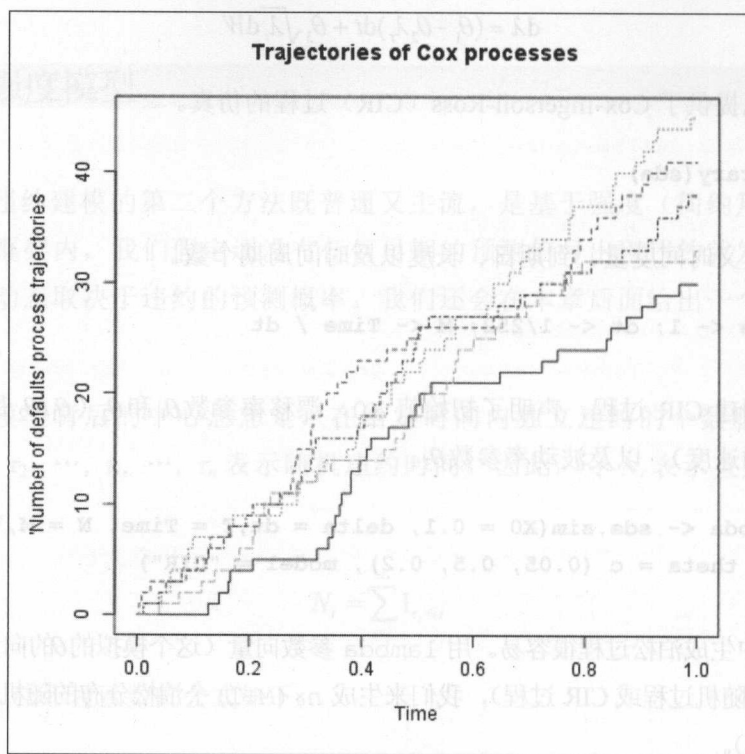


图 7-3 Cox 过程的轨线

7.2 相关违约——投资组合方法

在本节中，我们介绍如何使用相关随机变量 **copula** 处理信用组合的损失分布模拟。**copula** 函数是均匀分布的随机变量的联合累积分布函数。**copula** 函数包含了各部分的相依性结构的所有信息。任何连续分布的随机变量都可以转化成均匀分布的随机变量，这保证了一般性建模的可能性，比如，它可以和结构化方法结合在一起。使用 **copula** 包，我们演示了如何模拟两个服从 **Gaussian copula** 和 **t-copula** 分布的均匀随机变量，以及如何从生成的数据拟合 **Gaussian copula** 参数（也可以把这种方法运用到历史数据集上）。这个 **copula** 包可以为广泛主题提供有用的函数，比如对包括阿基米德 **copula** 在内的 **copula** 类提供作图和拟合函数。

首先，在载入 `copula` 包之后，声明一个相关系数为 0.7 的高斯 `copula` 类。

```
> library(copula)
> norm.cop <- normalCopula(0.7)
```

然后，生成两个随机变量的 500 个实现。这两个随机变量服从高斯 `copula` 相依性结构的均匀分布。

```
> set.seed(117); u1 <- rCopula(500, norm.cop)
```

为了比较，定义一个相关系数为 0.7 自由度为 4 的 `tcopula` 类。

```
> t.cop <- tCopula(0.7, df = 4)
```

现在，生成服从 `t-copula` 相依性的成对随机变量的 500 个实现。

```
> set.seed(117); u2 <- rCopula(500, t.cop)
```

把结果绘入相邻的两张图（见图 7-4）。`par` 命令确保两张图按照一行两列的位置方式（`mfcol`）相邻放置。

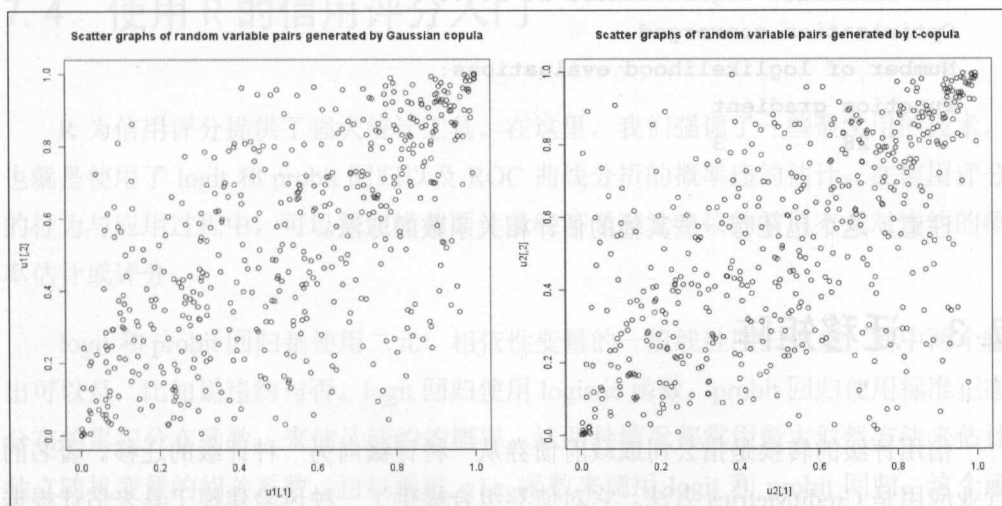


图 7-4 高斯 `copula` 和 `t-copula` 分别生成的随机变量对

```
> par(mfcol = c(1, 2))
> plot(u1, main = 'Scatter graphs of random variable pairs generated by
Gaussian copula')
> plot(u2, main = 'Scatter graphs of random variable pairs generated by
t-copula')
```

对 `u1` 数据使用极大似然方法 (ml) 拟合高斯 copula 的相关参数, 这个函数中我们使用了在前面定义的 `norm.cop` 对象的 copula 族类型。

```
> fit.ml <- fitCopula(norm.cop, u1, method = "ml")
```

当打印出下列结果, 我们发现估计的相关系数大约是 0.69。

```
> fit.ml
fitCopula() estimation based on 'maximum likelihood'
and a sample of size 500.
      Estimate Std. Error z value Pr(>|z|)
rho.1 0.68583 0.01936 35.43 <2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
The maximized loglikelihood is 164.8
Optimization converged
Number of loglikelihood evaluations:
function gradient
      28          3
```

注意, 这个包还有一些其他的估计相关系数的方法。

7.3 迁移矩阵

信用等级的转换是指公司或政府债券从一种评级向另一种评级的迁移。著名的行业应用是 CreditMetrics 方法。它对债券组合提供了一种风险建模工具来估计根据升降级引发的条件 VaR (Conditional Value-at-Risk, CVaR) 和组合信用价差。在本

节中，我们说明如何基于一个转移矩阵计算信用价差。

我们需要定义违约损失率 (loss given default, `lgd`)、评级 (在这个例子里是 A、B 和 D) 以及计算信用价差需要的一年转换矩阵。

```
> library(CreditMetrics)
> lgd <- 0.5
> rc <- c( "A", "B", "D")
> M <- matrix(c(85, 13, 2, 5, 80, 15, 0, 0, 100 ) /100, 3, 3,
+           dimnames = list(rc, rc), byrow = TRUE)
```

命令 `cm.cs` 用来从迁移矩阵计算信用价差。

```
> cm.cs(M, lgd)
      A      B
0.01005034 0.07796154
```

根据这个例子，从迁移矩阵 M 可以算出，一个评级为“A”的债务产品有大约 1% 的信用价差，而评定为“B”的债务大约有 7.8% 的信用价差。

7.4 使用 R 的信用评分入门

R 为信用评分提供了强大统计工具。在这里，我们强调了一些最常用的技术，也就是使用了 `logit` 和 `probit` 回归以及 ROC 曲线分析的概率违约估计。在信用评分的行为与应用过程中，可以按照横截面计量经济学理论建议的常用方式对违约的概率估计或评分。

`logit` 和 `probit` 回归是使用二元、相依性变量的一般线性回归模型，其中两个输出可以是，比如说违约与否。`logit` 回归使用 `logistic` 函数，`probit` 回归使用标准正态分布的累积分布函数，来估计违约的概率。这两种情况都常用最大似然方法来估计独立随机变量的相关系数。可以通过 `glm` 函数来调用 `logit` 和 `probit` 回归，这个函数在 R 中的功能是用来估计一般线性模型的系数。用于回归分析的典型 R 工具也适合进行深入检验。例如，`anova` 函数可以提供经典的方差分析。

信用评分模型经常使用受试者的操作特征曲线来说明它们所估计的模型表现。ROC 曲线表现了敏感性(敏感性, sensitivity, 肯定的非违约的对所有非违约的比率)对 1 减去特异性(特异性, specificity, 拒绝的违约对所有违约的比率)的比率。pROC 包包含了生成 ROC 曲线的 roc 函数。这个文档齐全的包可以按通常的方式进行安装并且可以使用 plot 命令画出 ROC 曲线。

7.5 小结

在本章中, 我们主要介绍了与信用风险建模有关的一些最常用方法。但是, 还有一些其他的行业方法可以用于处理违约风险。这些高级方法的基础往往是一些结构化方法或基于强度的方法。copula 模型是信用组合风险建模的主流方法, 尤其是在结构信用衍生品定价方面。R 中有多个易于理解的强大 R 包来对 copulas 建模。为降级风险建模的第一步是了解关于迁移矩阵管理和 CreditMetrics 方法的原则知识。最后, 我们简要叙述了 R 中信用评分的可行性。

其中, $F(x)$ 通常通过经验来估计。对于高于阈值的初始损失分布的尾部估计, 它代表了一个参数模型。

8.2 应用——保险理赔的建模

第 8 章 极值理论

极端损失的风险是金融领域和保险领域中许多风险管理问题的核心。一个极端的市场运动可能代表着某个投资者证券组合的显著下跌风险。未来信用损失的准备金大小必须覆盖发生极端风险时的贷款组合损失。一家银行需要的资本水平必须高到足够吸收极端的操作损失。保险公司需要准备好应付于自然产生或者人为造成的灾难, 甚至从未有过的数量级的灾难。

极值理论 (Extreme Value Theory, EVT) 与极端事件的统计分析有关。这种方法论提供的分布与极端观测一致, 同时它的参数形式受到理论的支持。EVT 的理论意义弥补了传统估计的不可靠性 (由极端事件的稀疏数据所引起)。EVT 允许量化极端事件的统计, 甚至超越了迄今大多数的极端观测。

EVT 在金融和保险中最有用的模型类型是阈值超标的模型。这些模型刻画了所有比较大的观测的分布, 这些观测都超过了某个高水平值, 因此提供了分布的尾部估计。因为许多风险管理问题都根据分布的尾部来进行公式化, 所以这些模型可以直接应用于这类问题。

本章的目的是通过一些真实世界的风险管理应用案例, 展示极值理论在金融和保险中可能使用的方法。首先, 我们提供了一个 EVT 中阈值超标模型的理论概览。然后, 我们详细解决一个例子, 用一个模型拟合火灾损失分布的尾部。我们使用拟合模型为火灾损失计算高分位数 (在险价值 VaR) 和条件分布 (Expected

Shortfall)。部分模型经常使用受试者的操作特征曲线来说明它们所估计的模型表现。

8.1 理论概览

令随机变量 X 表示我们需要建模的随机损失, $F(x)=P(X\leq x)$ 是它的分布函数。对于一个给定的阈值 u , 超过阈值 $Y=X-u$ 的超额损失的分布函数如下。

$$F_u(y) = P(X - u \leq y | X > u) = \frac{F(y+u) - F(u)}{1 - F(u)}$$

对一大类基础损失分布来说, 随着阈值向损失分布的右侧终点靠近, 超过一个高阈值 u 的超额损失的分布 $F_u(y)$ 会收敛到一般帕累托分布 (Generalized Pareto distribution, GPD)。这服从 EVT 中一个重要的极限定理。细节可以参考 McNeil、Frey 和 Embrechts (2005)。GPD 的累积分布函数如下。

$$G_{\xi, \beta}(y) = \begin{cases} 1 - (1 + xy/\beta)^{1/\xi}, & x \neq 0 \\ 1 - \exp(-y/\beta), & x = 0 \end{cases}$$

这里, ξ 一般是形状参数, 而 β 一般是尺度参数。

尽管严格来说, GPD 对超过一个高阈值的超额损失仅仅是一个极限分布。但是如果阈值有限, 它也可以当作超额损失的一个自然模型。换句话说, 如果阈值足够高, 那么可以认为超额分布已经足够接近 GPD, 因此后者可以用作超额分布的模型。首先, 我们假定对于某些 ξ 和 β 。

$$F_u(y) = G_{\xi, \beta}(y)$$

一旦一个 GDP 拟合到超额损失上, 它就可以对初始损失分布计算高分位数 (在险价值 VaR) 和条件期望。特别地, 可以对超过阈值 u 的损失分布函数建模如下。

$$F(x) = [1 - F(u)]G_{\xi, \beta}(x - u) + F(u)$$

其中, $F(u)$ 通常通过经验来估计。对于高于阈值的初始损失分布的尾部来说, 它代表了一个参数模型。

8.2 应用——保险理赔的建模

在本章的其余部分, 我们来解决一个在真实世界的风险管理应用中使用的 EVT 例子。我们将前面的方法运用于火灾保险理赔, 目的是对尾部拟合一个分布并提供分位数估计以及条件期望, 来刻画大型火灾损失的期望和量级。我们注意到, 完全相同的步骤也可以运用于信用损失或操作损失。对于市场风险的管理问题来说, 其中的基础数据通常是证券的回报率, 我们将在数据集中移除收益, 而只关注损失。否则, 建模步骤又成了重复。

R 中有多个包可以用于极端值分析。在本章中, 我们会在下面的命令里应用 `evir` 包。由 Gilleland、Ribatet 和 Stephenson (2013) 提供了一个关于 EVT 的多个 R 包的优秀概括。

正如前面所做, 使用之前需要安装并载入 `evir` 包。

```
> install.packages("evir")
> library(evir)
```

这个例子使用的数据取自丹麦的大型工业火灾保险理赔。这个数据集涵盖了从 1980~1990 年间, 损失超过 100 万丹麦克朗的所有火灾案例。这个数据集经常用于 EVT 的演示。`evir` 包提供了这个数据, 使用下面的命令我们可以把它载入工作空间。

```
> data(danish)
```

生成的数值向量包含了 2167 个观测以及对应的观测时间。输入 `help(danish)` 可以了解这个数据集的细节。

8.2.1 探索性数据分析

为了得到这个数据集一些直观印象，我们可以使用下列命令来计算和汇总统计量，并画出理赔的直方图。

```
> summary(danish)
Min. 1st Qu. Median Mean 3rd Qu. Max.
1.000 1.321 1.778 3.385 2.967 263.300
> hist(danish, breaks = 200, xlim = c(0,20))
```

图 8-1 展示了丹麦火灾保险理赔的直方图。

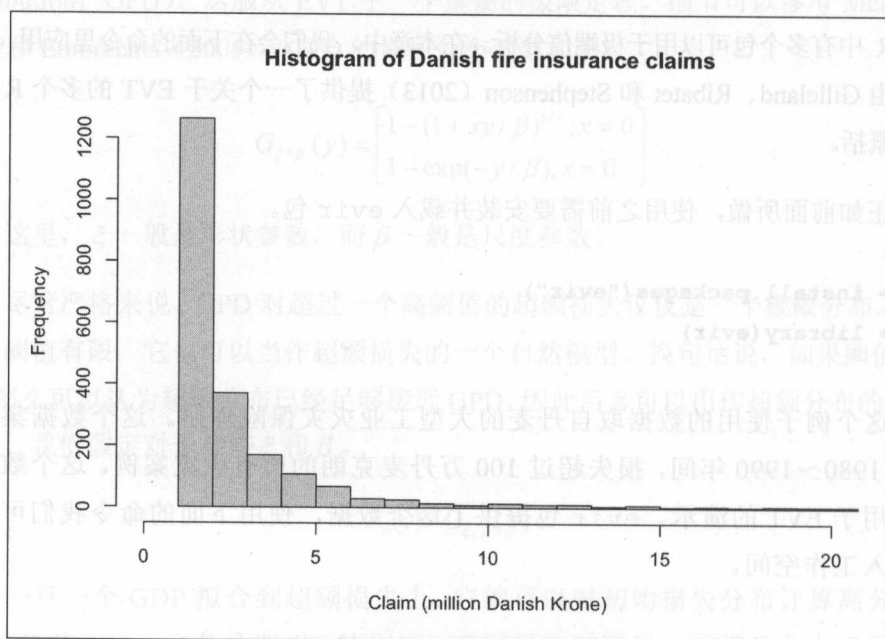


图 8-1 丹麦火灾保险理赔的直方图

理赔的分布严重有偏，右侧有一个长尾，这说明小型火灾损失发生的频率很高，但大型损失偶尔也会发生（数据集中最大的损失是 26300 万克朗）。这些大

型损失在图 8-1 中几乎看不到，因为我们已经在 2000 万克朗处截断了这张图。就像显示在下面命令行中的计算，高于这个阈值的损失小于 2%，但它们代表了总损失的 22%。

```
> sum(danish>20) / length(danish)
[1] 0.01661283
> sum(danish[danish>20]) / sum(danish)
[1] 0.2190771
```

在这个例子中，这就是我们感兴趣的那种极端大型损失发生的概率（以及它们预期的量级）。由于损失到如此规模的个数很少，因此不可能使用样本中大型损失的有关频率来估计概率。

8.2.2 理赔的尾部行为

另外一种数据可视化可能更有用，即对 x 轴（甚至是两个轴）取对数尺度。为了实现这个方法，我们可以通过使用 `evir` 包的 `emplot` 函数，画出经验互补累积分布函数（`ccdf`，即超过任何阈值的理赔的经验概率，有时也称为生存函数）。下面第一条命令创建了 x 轴使用对数尺度的命令，第二条命令产生了两个轴都采用对数尺度的图形。

```
> emplot(danish)
> emplot(danish, alog = "xy")
```

图 8-2 展示了上述第二张图。

有趣的是，当两个轴都采用对数尺度之后，画出的经验 `ccdf` 函数近似是线性的。这就解释了该数据的肥尾本质以及理赔可能是帕累托类分布（也被称为“幂律”，因为 `ccdf` 可被写为阈值的幂）。

分位数-分位数图（Q-Q 图）是检查数据是否来自于某种特定分布的另一种有用

工具。这种图形绘出了假设分布的分位数对比数据的分位数。如果分布的假定正确，那么生成的图形也是线性的。而图形非线性则说明数据的分布不同于假设的分布，例如，凹的图形说明经验分布的尾部更厚。

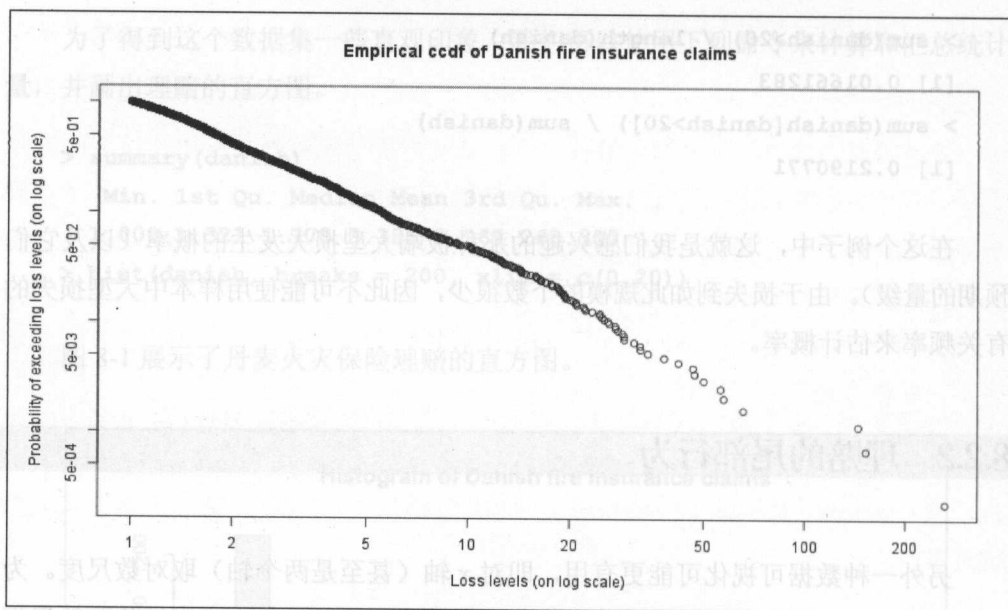


图 8-2 火灾保险理赔的经验 cdf 函数（双对数尺度）

Q-Q 图可以使用 `evir` 包的 `qplot` 函数创建。

损失数据的自然假设分布是指数分布。因此，`qplot` 函数比较了数据和缺省的指数分布。但是，这个函数通过参数 `xi` 指定形状参数 ξ ，允许数据和更一般的 GPD 分布进行比较。并且，通过指定 `trim` 参数，数据可以在某个值处右截尾，避免最大的那些观测值干扰图形。下面的命令创建了丹麦火灾损失数据对比指数分布的 Q-Q 图，其中损失数据在 100 处截尾。

```
> qplot(danish, trim = 100)
```

图形结果也确认了经验分布比指数分布尾部更厚，因此就这个数据来说，后者也不是一个好模型。

8.2.3 阈值的决定

现在已经确认了数据是厚尾的并且服从幂律，我们转向下一个问题，为阈值超标拟合一个 GPD 分布。但是，执行之前我们需要确定合适的阈值。尽管决定阈值是建模者的自主决断，但对于一个给定的阈值来说，已有一些有用的工具来帮助确认其对 GPD 的收敛是否足够充分。

对此可能最有用的工具是平均超额函数（mean excess function），定义为随机变量 X 超过阈值 u 的平均超额值。

$$e(u) = E(X - u | X > u)$$

很容易显示出，GPD 的平均超额函数是阈值 u 的线性函数，系数与分布的形状参数 ξ 成比例（因此一般来说，平均超额函数的正梯度表明了厚尾，同时负梯度表明了薄尾）。因此，一种决定阈值的合理方式是要找到这样一个值，超过它的样本平均超额函数是近似线性的。

evir 包的 meplot 函数画出了超过增加的阈值的样本平均超额。omit 参数允许你指定需要从图形上忽略的点数，这些点位于图形比较高的位置（这些点不会再次扰乱图形）。

```
> meplot(danish, omit = 4)
```

图 8-3 展示了生成的样本平均超额的图形。

图 8-3 表现出了样本的线性，并且穿过了损失的整个谱。因此，甚至可以对整个数据集拟合单个的 GPD 分布。但是，我们可以观察到，在刚好低于 10 的地方有一个小扭结，这表明更小的损失可能服从一个稍不同的法则。在 10 和 20 之间可以看到一个相当线性的区域。超过 20 之后，数据变得稀疏。因此，可以认为阈值为 10 是一个合理选择，这与样本平均超额函数相一致。剩下的超过这个阈值的观测有 109 个（原始样本的 5%）。

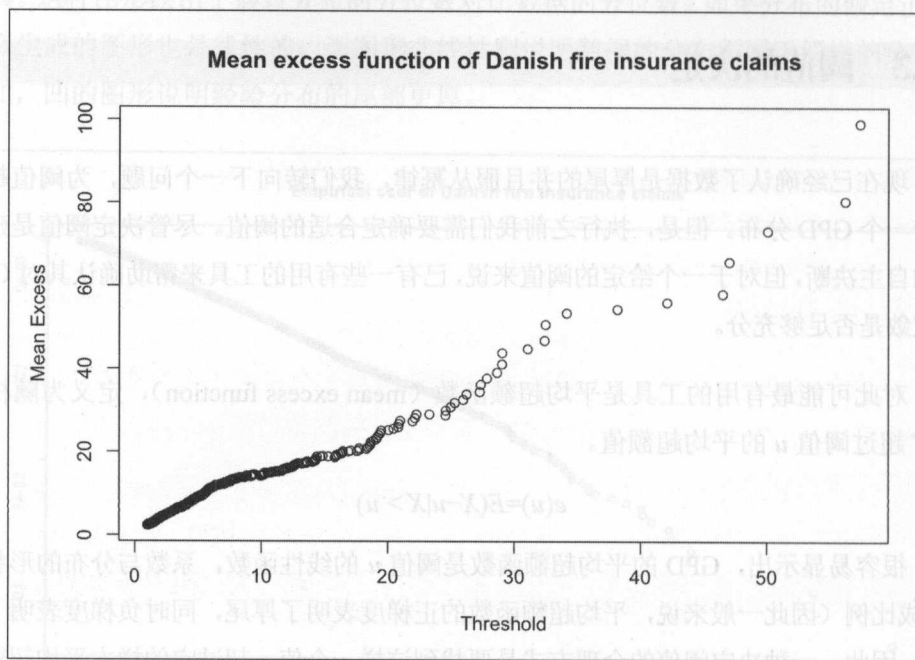


图 8-3 丹麦火灾保险理赔的平均超额函数

8.2.4 对尾部拟合 GPD 分布

现在万事具备，可以对火灾损失数据拟合一个 GPD 分布。使用下面的命令，我们可以使用 `gpd` 函数执行拟合，阈值指定如上节决定。

```
> gpdfit <- gpd(danish, threshold = 10)
```

缺省的 `gpd` 函数使用极大似然 (maximum likelihood, ML) 法来估计 GPD 分布的参数。这个函数返回一个 `gpd` 类的对象 (连同它们的标准误和协方差) 以及超过阈值的数据。`converged` 成分的零值说明使用极大似然估计时收敛到了最大值。成分 `par.ests` 和 `par.ses` 包含了估计的参数 ξ 和 β 以及它们各自的标准误。

```
> gpdfit$converged
```

```
[1] 0
```

```
> gpdfit$par.ests
```

```
xi      beta
```

```
0.4968062 6.9745523
```

```
> gpdfit$par.ses
```

```
xi      beta
```

```
0.1362093 1.1131016
```

于是, 极大似然估计得到估计参数是 $\xi=0.50$ 和 $\beta=6.97$, 各自的标准误是 0.14 和 1.11。

为了证明我们的结果, 我们会使用 `plot(gpdfit)` 命令, 它提供了一个菜单, 可以绘制超额的分布和原始分布的尾部 (连同拟合的 GPD), 以及拟合残差的散点图与 Q-Q 图。图 8-4 显示了超额分布和拟合的 GPD 分布 (来自菜单的 `plot1`), GPD 明显对数据提供了非常好的拟合。

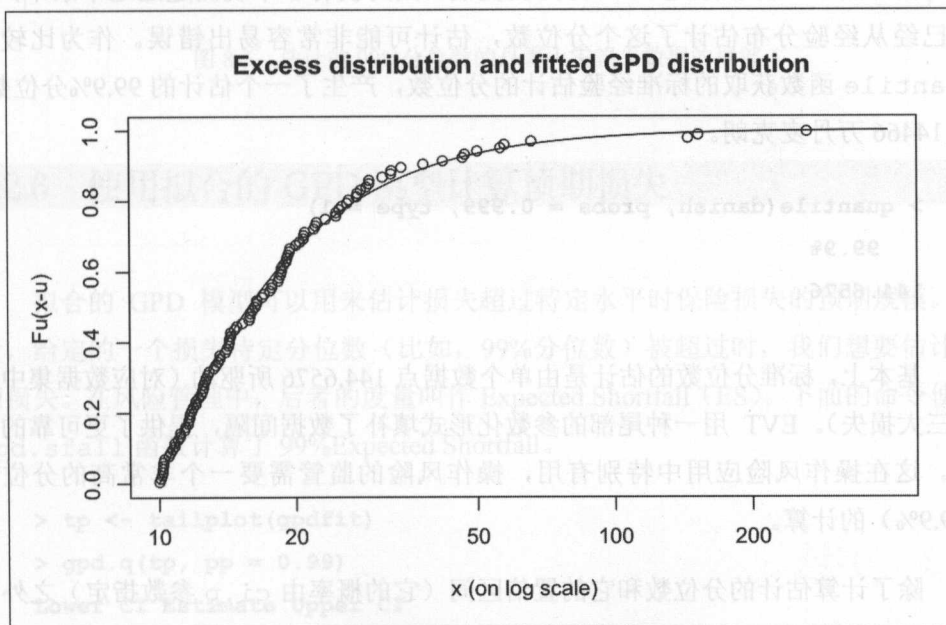


图 8-4 超额分布和拟合的 GPD 分布

8.2.5 使用拟合的 GPD 模型估计分位数

我们已经对数据拟合了一个 GPD 模型，现在可以使用它来估计高分位数或者 VaR。我们可以使用 `gpd.q` 函数来实现，但这个函数需要 `plot.gpd` 函数或者 `tailplot` 函数（对应于 `plot.gpd` 的 `selection 2`）返回的列表对象。我们使用 `tailplot` 函数直接创建一个原始的丹麦火灾损失分布的尾部图形。然后把返回的对象传递给 `gpd.q`，并把参数 `pp` 指定为要估计的分位数。

```
> tp <- tailplot(gpdfit)
> gpd.q(tp, pp = 0.999, ci.p = 0.95)
Lower CI Estimate Upper CI
64.66184 94.28956 188.91752
```

估计的 99%分位数是 9429 万丹麦克朗。因为仅有 3 个观测超过这个水平，我们已经从经验分布估计了这个分位数，估计可能非常容易出错误。作为比较，`quantile` 函数获取的标准经验估计的分位数，产生了一个估计的 99.9%分位数，是 14466 万丹麦克朗。

```
> quantile(danish, probs = 0.999, type = 1)
99.9%
144.6576
```

基本上，标准分位数的估计是由单个数据点 144.6576 所驱动（对应数据集集中的第三大损失）。EVT 用一种尾部的参数化形式填补了数据间隔，提供了更可靠的估计。这在操作风险应用中特别有用，操作风险的监管需要一个非常高的分位数（99.9%）的计算。

除了计算估计的分位数和它的置信区间（它的概率由 `ci.p` 参数指定）之外，`gpd.q` 函数还对 `tailplot` 生成的图形加上了一条轮廓线，显示了分位数的点估计（垂直的虚线）以及估计量的轮廓似然曲线（虚曲线）。估计的置信区间边界由虚

曲线和水平虚直线的交点给出，显示在图 8-5 中。

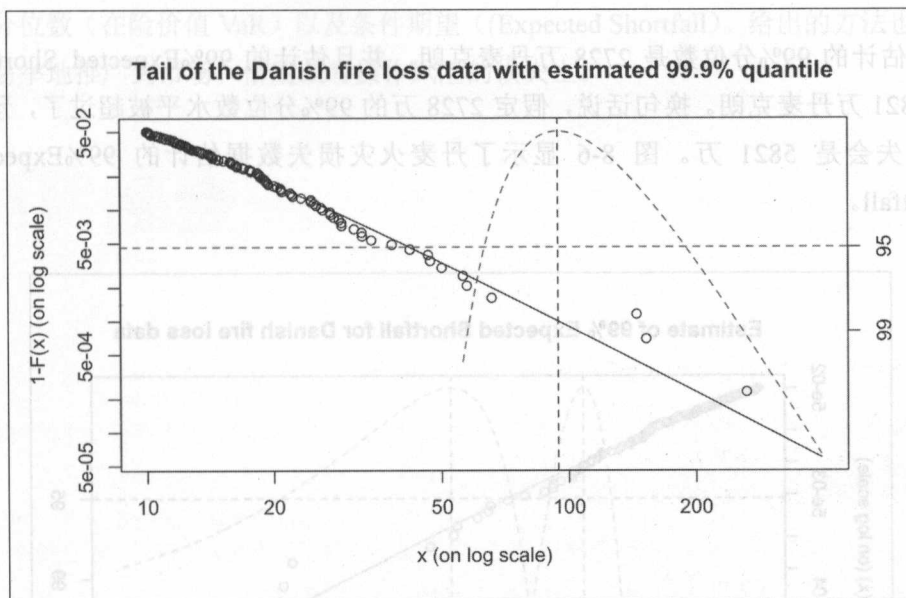


图 8-5 估计的 99%分位数的丹麦火灾损失数据的尾部

8.2.6 使用拟合的 GPD 模型计算预期损失

拟合的 GPD 模型可以用来估计损失超过特定水平时保险损失的预期规模。或者，给定的一个损失特定分位数（比如，99%分位数）被超过时，我们想要估计预期损失。在风险管理中，后者的度量叫作 Expected Shortfall (ES)。下面的命令使用 `gpd.sfall` 函数计算了 99% Expected Shortfall。

```
> tp <- tailplot(gpdfit)
> gpd.q(tp, pp = 0.99)
Lower CI Estimate Upper CI
23.36194 27.28488 33.16277
> gpd.sfall(tp, 0.99)
```


Lower CI Estimate Upper CI
41.21246 58.21091 154.88988

估计的 99%分位数是 2728 万丹麦克朗，并且估计的 99%Expected Shortfall 是 5821 万丹麦克朗。换句话说，假定 2728 万的 99%分位数水平被超过了，预期的损失会是 5821 万。图 8-6 显示了丹麦火灾损失数据估计的 99%Expected Shortfall。

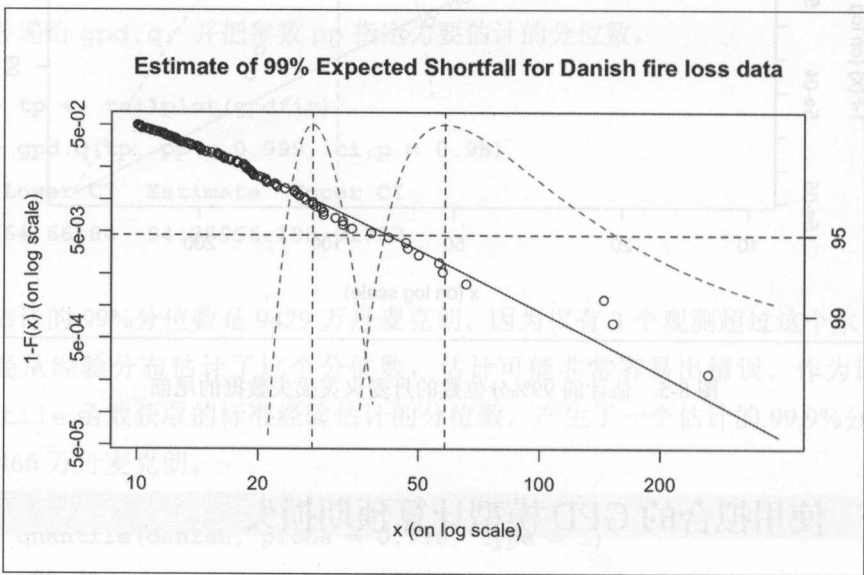


图 8-6 丹麦火灾损失数据 99%Expected Shortfall 的估计

生成的图 8-5 同时显示了 99%分位数或者 VaR（第一条垂直虚线和对应的轮廓似然曲线）和 99%Expected Shortfall（第二条垂直虚线和对应的轮廓似然曲线）。

8.3 小结

在本章中，我们给出了一个如何使用 R 将极端值理论应用在真实世界风险管理

中的案例研究。在简要概括了 EVT 中阈值超标模型的理论之后,我们解决了一个对保险理赔的分布尾部拟合模型的详细案例。我们使用拟合的模型对火灾损失计算了高分位数(在险价值 VaR)以及条件期望(Expected Shortfall)。给出的方法也能轻而易举地推广到市场、信用或者操作风险的损失上。

网络性质

网络	节点	边/路径长度	度分布
规则(比如环、全网络)	高	高	在每一个节点相等或具有固定的进/出度
随机机(比如 Erdős-Rényi)	低	低	指数、高斯
小世界(比如 Barabási-Albert)	低	低	幂律、高斯

这张表格的来源是 Markose et al. 2009。

网络性质如下:

• 密度度量了节点之间的连接程度。密度越高,网络越稠密。

人们认为它是社交网络的标志。整个网络的平均度是衡量网络的一个重要指标。

• 平均度是网络中所有节点的度的平均值。

• 路径长度是两个代理之间的距离,由分割它们的边个数给出。最短路径的平均值是网络的特征。网络中的最长或最短路径称作直径。

• 度是一个节点对其他节点的连接个数。度分布是网络中所有节点的度的分布。

图 8-6 显示了丹麦火灾损失数据估计的 99% 分位数。估计的 99% 分位数是 2728 万丹麦克朗。如果假设 99% 分位数是 5821 万丹麦克朗。换句话说，假定 2728 万的 99% 分位数，的损失会是 5821 万。图 8-6 显示了丹麦火灾损失数据估计的 Shortfall。

第 9 章 金融网络

Estimate of 99% Expected Shortfall for Danish fire loss data

在前一章中，我们已经看到对服从非对称且厚尾的分布的极端事件如何建模，以及与这些极端事件相联系的风险如何度量和和管理。

在某些情况下，我们可以利用可得的金融数据来构建复杂网络。在金融网络中，某些属性（度、质量等等）的分布高度不对称并且是厚尾的。

可得的金融网络究其本质并不完整。它们既没包含所有的参与者，也没包含所有的联系或所有的相关属性。但即使状态有所限制，它们仍包含了极其丰富详实的数据集，可以帮助我们深入了解研究市场的细致微观结构。

关于金融网络如何在 R 中表示、模拟、可视化以及分析，本章提供了一个概览。我们关心两个重要的应用问题。

- 如何发现网络拓扑结构的改变。
- 如何借助于中心性度量识别系统性重要的参与者。

9.1 金融网络的表示、模拟和可视化

网络可以通过成对的列表、邻接矩阵或者图来表示。图包括顶点（节点）和边。在 R 中，顶点被编号并可以有多个属性。两个顶点之间可以存在一条边（有向或无

向的、加权或不加权的), 并且边也可以有其他的属性。在大部分金融网络中, 顶点代表市场参与者, 而边描述了他们之间不同的金融联系。

使用内置的 R 工具以及 `igraph` 包的某些函数, 很容易可以创建、模拟人工网络。表 9-1 汇总了一些重要的网络类型及其基本性质。

表 9-1

网络的性质

网络	聚类	平均路径长度	度分布
规则 (比如环、全网络)	高	高	在每一个节点相等的或者固定的进出度
纯随机 (比如 Erdős-Rényi)	低	低	指数、高斯
无标度	变量	变量	幂律/厚尾
小世界 (比如 Barabási, Watz-Strogatz)	高	低	幂律/厚尾

这张表格的来源是 Markose at al. 2009。

最重要的网络性质如下。

- 密度度量了节点之间链接的程度, 关于一个完全图中所有可能的链接。
- 聚类 (在 R 中叫作传递性) 度量了每个代理 (agent) 的邻居如何相互连接, 人们认为它是社交网络的标志。整个网络的聚类系数是它的节点的所有系数均值。
- 路径长度是两个代理之间的距离, 由分割它们的边个数给出。最短路径的均值是整个网络的特征。网络中的最长最短路径叫作直径。
- 度是一个节点对其他节点的连接个数。度分布是整个网络之上这些度的概率分布。

非随机规则网络高度有序, 其中每个节点与它所有的近邻节点相连接。一个完整的图可以使用 `igraph` 包的 `graph.full` 函数生成, 部分网络可以使用相似的

函数生成树、网格、环等。

与规则网络相反，在一个纯随机的 Erdős-Rényi 网络中，通过均匀随机地选择两个节点来生成链接。因为此时我们在处理随机数，所以需要设置自定义的随机种子以及随机数生成器的状态，以便所有的 R 会话可以返回相同的随机数。

```
> set.seed(7)
```

在仿真一个 Erdős-Rényi 图时，我们需要预先设置至少两个参数：节点个数（比如，100）以及在两个任意节点之间画出一条边的概率（比如，0.1）。

```
> e <- erdos.renyi.game(100, 0.1)
```

```
> plot(e)
```

图 9-1 描绘了一个纯随机网络（Erdős-Rényi）。

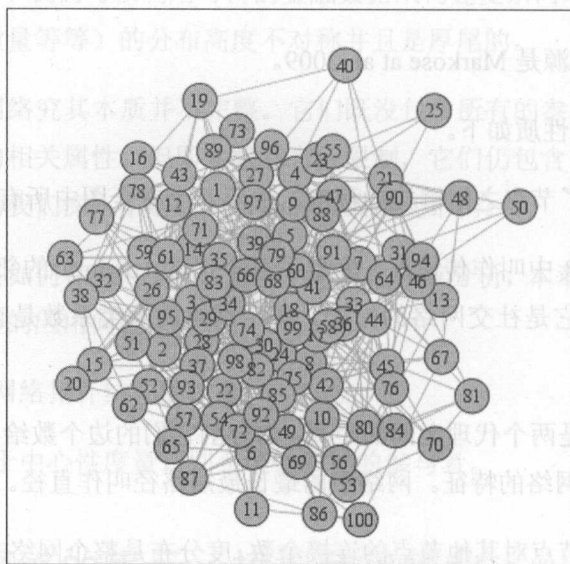


图 9-1 Erdős-Rényi 网络

我们也可以计算出上述网络的主要特征，它们是表 9-1 中的密度、聚类（传递性）以及平均路径长度。


```

> graph.density(e)
[1] 0.05434343
> transitivity(e)
[1] 0.05522828
> average.path.length(e)
[1] 2.923636

```

图密度和传递性大约为 0.1，初始设置的边密度和差异仅仅源于小样本固有的噪声。

在一个无标度网络中，度分布服从幂律，因此各顶点的度差异相当大。小世界网络构成了无标度网络的一个特定子集，其中的顶点趋向于形成团体 (cliques)，结果导致了弱链接枢纽的过剩。毫不奇怪，在小世界网络中，聚合系数明显高而平均路径短。为了解释度分布的幂律和聚合，有人提出了偏好连接和拟合度的机制，参见 Barabási-Albert (1999) 和 Bianconi-Barabási (2001)。社会/金融网络通常作为小世界来建模。在 R 中有多种方法可以创建小世界网络，比如 `watts.strogatz.game` 和 `barabasi.game`。在这里，我们使用第一个。

```

> set.seed(592)
> w <- watts.strogatz.game(1, 100, 5, 0.05)
> plot(w)

```

图 9-2 描绘了一个随机的无标度网络 (Watts-Strogatz)。

根据主要的网络度量，我们来比较 Watts-Strogatz 小世界网络和之前的纯随机 Erdős-Rényi 图。

```

> graph.density(w)
[1] 0.1010101 # 近似相同
> transitivity(w)
[1] 0.4626506 # 更高
> average.path.length(w)
[1] 2.625455 # 更长

```

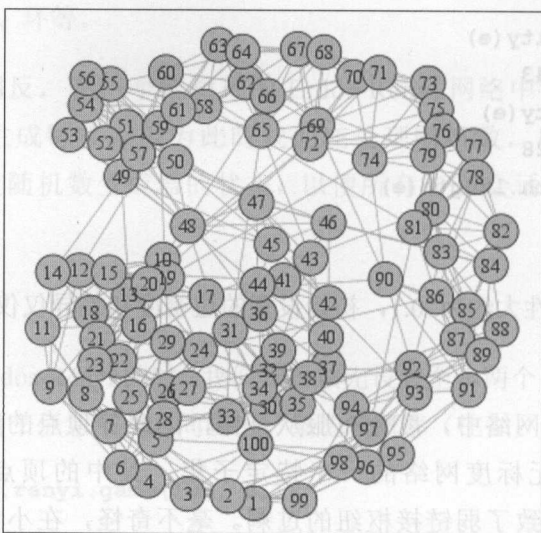


图 9-2 无标度网络

在上面的 R 代码块中，我们把 Watts-Strogatz 游戏的结果存储在一个叫作 `w` 的变量中，之后再把它画出来。使用 `get.edgelist` 函数可以很容易地提取边的列表，这个函数会返回一个两列的矩阵。这里我们仅显示返回列表的前 5 行。

```
> we <- get.edgelist(w) > head(we, 5)
```

	[,1]	[,2]
[1,]	1	2
[2,]	2	77
[3,]	3	4
[4,]	4	5
[5,]	5	6

其他的网络操作在 R 中也可以完成。比如，我们可以借助于 `get.adjacency` 函数来见到图的邻接矩阵。或者，通过使用 `permute.vertices` 改换顶点的 ID 序列，可以帮助随机化网络。如果需要连同顶点属性将多个顶点合并为一个，可以使用 `contract.vertices`。使用一些据此命名的 R 函数，我们还能创建几个网络的 `union` 和 `intersection`。

9.2 网络结构的分析和拓扑改变的检查

现在，假定我们已经获得了一个真实世界的银行间市场的数据库（为了演示，我们给出随机化的数据），其中的银行相互借贷，并在 2007 年~2010 年期间的每天结束时，贷款银行出具他们的头寸报告。这个数据库包含了 50 家银行的数据，并且贷款期限为一天。为了在 R 中操作一个真实世界的网络，可以把我们的数据转换为一个 CSV 文件并把它保存在工作目录中。表 9-2 显示了这个 CSV 文件的开头几行。

表 9-2 一个银行间市场的数据库

银行	合作方	数量	利息	年	月	日
1	21	5	7.9	2007	1	3
1	42	3	7.9	2007	1	3
10	11	0.35	7.8	2007	1	3
18	24	2	8	2007	1	3
2	11	1.3	7.8	2007	1	3
21	11	0.8	7.8	2007	1	3
21	2	5	7.75	2007	1	3
3	24	4	7.95	2007	1	3

表 9-2 的每一行包含一笔交易：报告银行（贷方）、合作银行（借方）、贷款数量、利率以及交易的日期。我们可以从 CSV 文件中读到这些数据的细节。

```
> data <- read.csv2('networktable.csv')
```

现在，我们有一个 21314 行 7 列的表格。

```
> str(data)
'data.frame': 21314 obs. of 7 variables:
 $ Bank      : int 1 1 10 18 2 21 21 3 3 30 ...
 $ Partner   : int 21 42 11 24 11 11 2 24 42 12 ...
 $ Amount    : num 5 3 0.35 2 1.3 0.8 5 4 1.8 2 ...
 $ Interest  : num 7.9 7.9 7.8 8 7.8 7.8 7.75 7.95 7.9 7.9 ...
 $ Year      : int 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
 $ Month     : int 1 1 1 1 1 1 1 1 1 1 ...
 $ Day       : int 3 3 3 3 3 3 3 3 3 3 ...
```

每家银行的资产负债表规模也可以得到，存储在一个分开的 CSV 文件中。这个 CSV 文件的第一列假定包含了象征性的顶点名称，其他各列作为附加的顶点属性加入。

根据下面的方式，我们可以使用数据框创建图对象。

```
> size <- read.csv2('vertices.csv')
```

这个函数创建了一个 igraph 对象。数据框必须在前两列中包含边的列表。额外的列可以看作是边的属性。

```
> bignetwork <- graph.data.frame(data, vertices = size)
```

在第二步，我们可以请求网络的基本性质。

```
> is.connected(bignetwork)
```

```
[1] TRUE
```

这个网络是充分连接的，意味着在 2007 年~2010 年间每家银行都和其他所有银行至少交易了一次。我们可以检查这个网络的边是否是多重的。

```
> table(is.multiple(bignetwork))
```

```
FALSE TRUE
```

```
1049 20265
```

R 发现许多边是多重的，这意味着银行们和同一家合作银行交易了不止一次。我们还要检查这个网络是否存在环，即，交易的报告银行与合作银行是否为

同一家。

```
> str(is.loop(bignetwork))
logi [1:21314] FALSE FALSE FALSE FALSE FALSE FALSE ...
```

在这里，我们得到了所有边的列表。看起来有一些环必然伴随着数据的错误。如果希望去掉所有的环并汇总多重边，我们可以通过简化网络一步实现这个任务。

```
> snetwork <- simplify(bignetwork, edge.attr.comb = "sum")
```

图形简化后，我们用相对小的箭头把它画出来，使结果图不要太拥挤。

```
> plot(snetwork, edge.arrow.size = 0.4)
```

图 9-3 描绘了真实世界的网络。

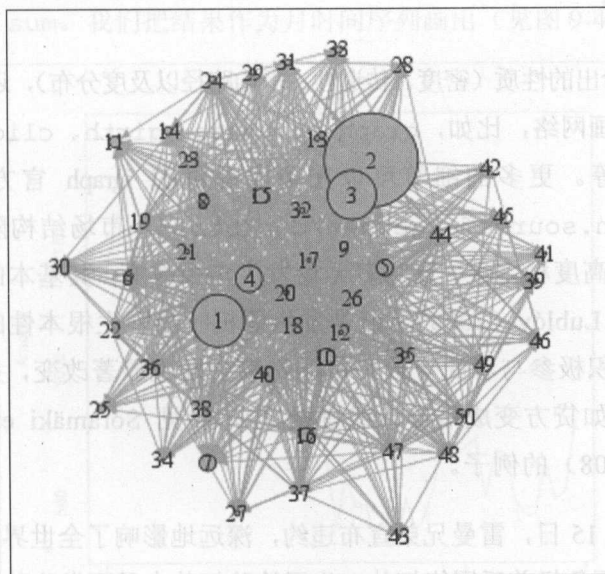


图 9-3 真实世界的银行借贷网络

在第三步，我们运用内置在 R 中的复杂分析工具，来探索网络的内部结构。

我们可以检测出社区（即紧密连接的子图），例如，根据 `walktrap.community` 函数找到紧密连接的子图。其思想是短的随机游走总是保持在一个相同的社区内。

如果边是随机分布的，那么模块度是指属于给定组的边部分减去预期的部分。模块度的范围是 $[-1/2, 1]$ 。它如果为正，就是聚合的标志。

```
> communities
Number of communities (best split): 2
Modularity (best split): 0.02499471
Membership vector:
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  2  1  2  2
47 48 49 50
 2  2  2  2
```

除了表 9-1 给出的性质（密度、传递性、平均路径以及度分布），还有很多其他的 R 函数也能用来刻画网络，比如，`graph.coreness`、`girth`、`cliques.number`、`reciprocity` 等。更多的细节可以在下面提供的 `igraph` 官方手册中找到：<http://igraph.sourceforge.net/doc/html/>。市场结构随着时间变动。它们平时表现出高度稳定性，意味着无论发生多少改变，其基本的拓扑结构保持不变，比如参见 Lublóy (2006)。但是发生危机时会发生根本性的变动：市场的枯竭和再发展，积极参与者的个数和市场份额会发生显著改变，并且参与者的角色也会改变（比如贷方变成借方，或者相反），参见 Soramäki et al. (2006) 和 Bech-Atalay (2008) 的例子。

2008 年 9 月 15 日，雷曼兄弟宣布违约，深远地影响了全世界的金融市场。因此，应该比较一下危机前后网络拓扑。为了检验拓扑中是否发生基本改变，我们来创建一个聚合网络的月度序列，然后计算每个月的网络度量并把它们作为时间序列画出来。

为了仅匹配一部分数据，我们可以使用 `subset` 函数。比如，为了滤出 2008 年 9 月的数据，我们可以这样运行。

```
> monthlynetwork <- subset(data, (Year == 2008) & (Month == 9))
```

在接下来的一些例子里，我们会迭代选取原始数据集中一个月的子集，并将一些函数作用在这些子集上。可以通过一个基本循环的方法来实现，结合不同的 `apply` 函数（特别是来自 `plyr` 包的 `ddply` 函数），或者基于给定的维度聚集数据集。我们始于聚集度量，并逐渐展开细节。因此，我们先来看看，聚集量（一个月內贷款数量的和）如何随着时间变动。

```
> mAmount <- with(data,
+   aggregate(Amount, by = list(Month = Month, Year = Year),
+   FUN = sum))
```

在这里，我们借助于 `aggregate` 函数，计算了 `data` 内每一个 `Year` 和 `Month` 的 `Amount` 的和 `sum`。我们把结果作为月时间序列画出（见图 9-4），命令如下。

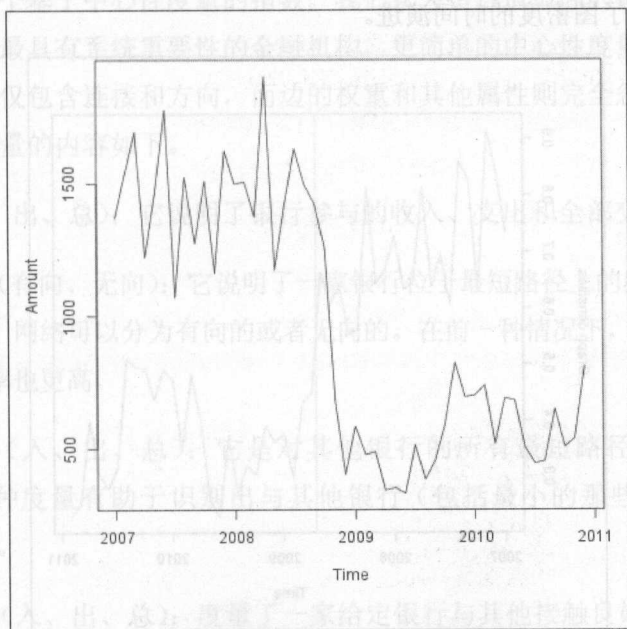


图 9-4 月度贷款数量的变化

```

800 > plot(ts(mAmount$x, start = c(2007, 1), frequency = 12),
+       ylab = 'Amount')

```

图 9-4 描绘了月度数量随时间的变化。

在图 9-4 上, 我们可以观察到网络在雷曼刚刚倒掉之后发生了一个重要的结构性改变。如果更仔细地观察这个网络, 我们会发现网络的内部结构在危机爆发后已经发生显著的改变。为了证实这些改变, 我们一个月一个月地计算出网络度量的时间序列并绘出。我们使用嵌套的循环, 计算每个月的图密度值。

```

> ds <- sapply(2007:2010, function(year) {
+   sapply(1:12, function(month) {
+     mdata <- subset(data, (Year == year) & (Month == month))
+     graph.density(graph.data.frame(mdata))
+   })
+ })
> plot(ts(as.vector(ds), start = c(2007, 1), frequency=12))
> abline(v = 2008 + 259/366, col = 'red')

```

图 9-5 描绘了图密度的时间演进。

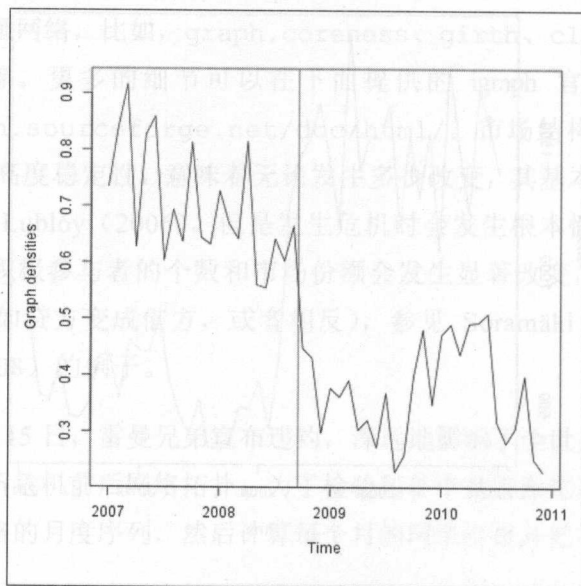


图 9-5 图密度的变化

我们可以看到,雷曼倒闭后,网络密度突然下降,这反映了交易集中在更少的银行中。大部分其他的网络度量也表现出显著的结构改变。

9.3 对系统风险的贡献——系统重要性金融机构的识别

一个复杂系统并非其各个元素的简单加总。有可能所有实体自身都很安全,但作为整体的系统却依然脆弱。系统性风险是由一个或多个冲击导致系统崩溃的风险。如果需要识别 BCBS (2011)提出的系统重要性金融机构(systemically important financial institutions, SIFIs),我们需要考虑导致系统性风险的5个因素:规模、内部关联性、缺乏替代物、交叉管辖的行为,以及行为的复杂性。当度量内部关联性时,我们可以依据网络数据并运用多种方法,比如,中心性度量、压力测试以及核心-边缘模型。

现在,基于 Komárková et al. (2012) 和 von Peter (2007) 的描述,我们来说明第一种方法,一个基于中心性度量的指数。我们认为指数值最高的银行是最中心的银行,因此也是最具有系统重要性的金融机构。更简单的中心性度量则基于更小的信息集,其中仅仅包含连接和方向,而边的权重和其他属性则完全忽略。比如,更简单的中心性度量的内容如下。

- 度(入、出、总):它说明了银行参与的收入、支出和全部交易的总数。
- 中间性(有向、无向):它说明了一家银行位于最短路径上的频率。决定最短路径时,网络可以分为有向的或者无向的。在前一种情况下,最短路径更长,因此频率也更高。
- 紧密性(入、出、总):它是对其他银行的所有最短路径的倒数的平均值。这种度量有助于识别出与其他银行(包括最小的那些)接触最广泛的银行。
- 特征值(入、出、总):度量了一家给定银行与其他接触良好的银行的联系程度。

我们取 2007 年到 2010 年作为一个整体，关注一个有向图的所有连接，使用的命令如下。

```
> g <- graph.data.frame(data)
```

我们计算了 4 个中心性度量，并加总到一个指数中。

```
> degree <- degree(g, normalized = TRUE)
> between <- betweenness(g, normalized = TRUE)
> closeness <- closeness(g, normalized = TRUE)
> eigenv <- evcent(g, directed = TRUE)$vector
```

当计算这个指数时，对一家给定银行的中心性度量与整个总体的中心性度量均值之差，我们必须标准化。为了实现这个目的，如果还不想使用有多个参数的内置 scale 函数，我们需要构造一个函数。

```
> norm <- function(x) x / mean(x)
```

如果中心性度量是等权重的，我们可以使用下面的公式。

```
> index <- (norm(degree) + norm(between) +
+ norm(closeness) + norm(eigenv)) / 4
> index
```

1	8	15	2	18	3
0.91643667	0.49431153	1.06216769	1.35739158	4.56473014	1.44833480
26	32	35	36	37	4
1.36048296	0.73206790	1.13569863	0.40296085	0.54702230	3.94819802

现在，我们画出指数的分布并选择指数值最高的银行，如图 9-6 所示，可以使用下面的生成命令。

```
> hist(index)
```

我们一旦有了这个指数的分布，就必须决定考虑的银行中哪一家是重要的（比如，

那些指数值高于 2.5 的银行：18、12、20、17、9 和 10)。当然，这个指数可以通过其他的中心性度量来完成，即，基于更广泛信息集的更高级指数，参见 von Peter (2007) 介绍的例子。

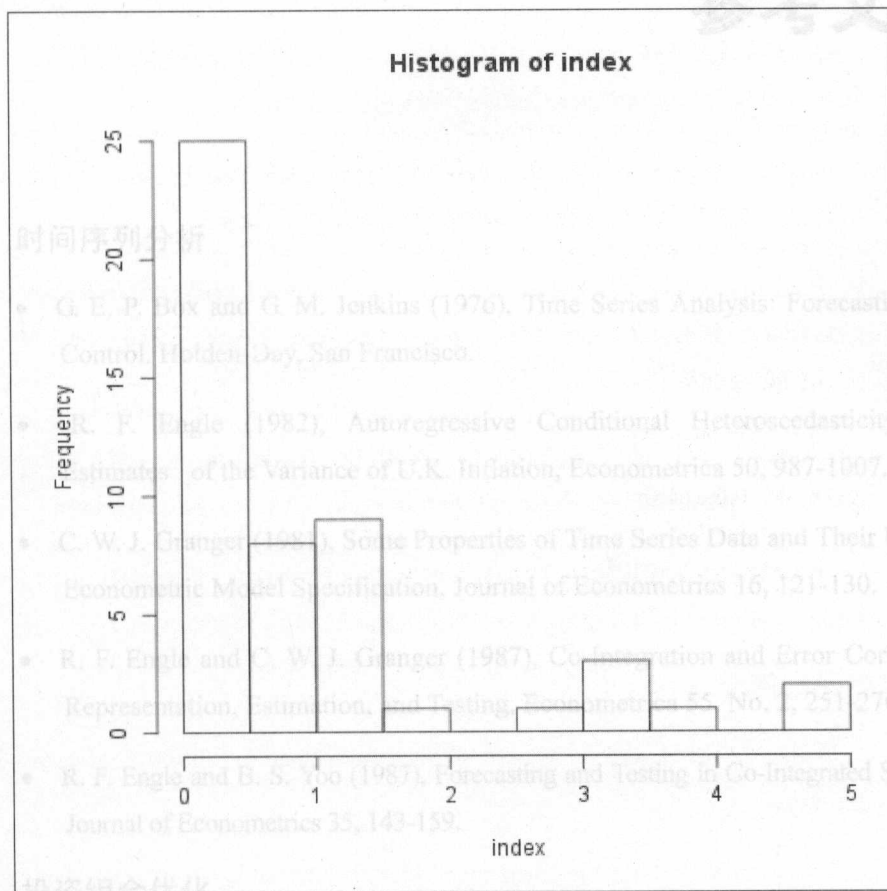


图 9-6 中心性度量指数的分布

9.4 小结

在本章中，我们关注金融网络和 R 包 `igraph` 的使用，这个包提供了有效工

具用于网络的模拟、操作、可视化以及分析。我们学习了如何读入网络数据以及如何探索网络的基本性质。我们揭示了案例市场数据由于金融危机存在着显著的结构改变。在最后一部分，我们给出了一种简单方法可以找到网络中系统重要性的金融机构。

我们计算了4个中心性指标来衡量每个指数中。

```
> degree <- degree(g, normalized = TRUE)
> between <- betweenness(g, normalized = TRUE)
> closeness <- closeness(g, normalized = TRUE)
> eigenv <- evcent(g, directed = TRUE)$vector
```

在计算这个指数时，对一家给定银行的中心性度量与整个总体的中心性度量均值之比，我们必须标准化。为了实现这个目的，如果我们不想使用内置的内置函数，我们需要构造一个函数。

```
> norm <- function(x) x / mean(x)
```

如果中心性度量是等权重的，我们可以使用下面的公式。

```
> index <- (norm(degree) + norm(between) +
+ norm(closeness) + norm(eigenv)) / 4
> index
```



index	frequency
0.0	1
0.25	2
0.5	3
0.75	4
1.0	5
1.25	6
1.5	15
1.75	10
2.0	8
2.25	5
2.5	3
2.75	2
3.0	1

该索引度量是网络中心性的度量。

最后，我们画出所算的分布并选择指数值最高的银行，如图9-6所示，可以使用下面的代码命令。

```
> hist(index)
```

图9-6

我们，只看了这个指数的分布，就必须决定要选哪些银行。这是困难的（比如，我们选了哪些银行，用哪个银行来衡量网络中心性，等等）。本章本

参考文献

时间序列分析

- G. E. P. Box and G. M. Jenkins (1976), Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco.
- R. F. Engle (1982), Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of U.K. Inflation, *Econometrica* 50, 987-1007.
- C. W. J. Granger (1981), Some Properties of Time Series Data and Their Use in Econometric Model Specification, *Journal of Econometrics* 16, 121-130.
- R. F. Engle and C. W. J. Granger (1987), Co-Integration and Error Correction: Representation, Estimation, and Testing, *Econometrica* 55, No. 2, 251-276.
- R. F. Engle and B. S. Yoo (1987), Forecasting and Testing in Co-Integrated Systems, *Journal of Econometrics* 35, 143-159.

投资组合优化

- P. Carl and B.G. Peterson (2013), PerformanceAnalytics: Econometric Tools for Performance and Risk Analysis. Available at <http://cran.r-project.org/package=PerformanceAnalytics>.
- R. McTaggart and G. Daróczy (2013), Quandl: Quandl Data Connection. Available at <http://cran.r-project.org/package=Quandl>.

- R. C. Merton (1993), On the Microeconomic Theory of Investment under Uncertainty, Handbook of Mathematical Economics, in: K. J. Arrow and M.D. Intriligator (ed.), Handbook of Mathematical Economics, edition 4, volume 2, chapter 13, 601-669, Elsevier.
- J.A. Ryan (2013), quantmod: Quantitative Financial Modelling Framework. Available at <http://cran.r-project.org/package=quantmod>.
- A. Trapletti and K. Hornik (2013), tseries: Time Series Analysis and Computational Finance. Available at <http://cran.r-project.org/package=tseries>.
- W. F. Sharpe (1964), Capital Asset Prices: A Theory of Market Equilibrium under Conditions Of Risk, Journal of Finance, American Finance Association 19, No. 3, 425-442, 09.
- D. Wuertz and M. Hanf (2010), Portfolio Optimization with R/Rmetrics (Rmetrics Association & Finance Online) Available at www.rmetrics.org.
- D. Wuertz and Y. Chalabi (2013), timeSeries: Rmetrics - Financial Time Series Objects. Available at <http://cran.r-project.org/package=timeSeries>.

资产定价模型

- Z. Bodie, A. Kane, and A. Marcus (2004), Investments, Edition 6, McGraw-Hill Irwin.
- J. H. Cochrane (2005), Asset pricing, Princeton University Press, New Jersey.
- J. Lintner (1965), Security Prices, Risk, and Maximal Gains from Diversification, Journal of Finance 20, No. 4, 587-615.
- J. Lintner (1965), The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budget, Review of Economics and Statistics 47, No. 1, 13-37.

- P. Medvegyev and J. Száz (2010), A meglepetések jellege a pénzügyi piacokon. Bankárképző, Budapest.
- M. Miller and M. Scholes (1972), Rates of Return in Relation to Risk: A Re-examination of Some Recent Findings, in: Studies in the Theory of Capital Markets, New York, Praeger, 47-78.
- S. A. Ross (1976), Return, Risk and Arbitrage, in: Risk and Return in Finance, Cambridge, Mass, Ballinger.
- W. F. Sharpe (1964), Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk, Journal of Finance 19, No. 3, 425-442.
- P. Wilmott (2007), Paul Wilmott Introduces Quantitative Finance, Edition 2, John Wiley & Sons Ltd, West Sussex.

固定收益证券

- J. C. Hull (2012), Options, Futures, and Other Derivatives, 8th edition, Prentice Hall.
- Z. Bodie, A. Kane, and A. J. Marcus (2008), Investments, 8th edition, McGraw-Hill.
- K. G. Nyborg (1996), The Use and Pricing of Convertible Bonds, Applied Mathematical Finance 3, No. 3, 167-190.
- M. J. Brennan and E. S. Schwartz (1980), Analyzing Convertible Bonds, Journal of Financial and Quantitative Analysis 15, 907-929. DOI: 10.2307/2330567.

估计利率期限结构

- J. H. McCulloch (1971), Measuring the Term Structure of Interest Rates, The Journal of Business 44, 19-31.
- J. H. McCulloch (1975), The Tax-Adjusted Yield Curve, The Journal of Finance 30,

811-830.

- R. Ferstl and J. Hayden (2010), Zero-Coupon Yield Curve Estimation with the Package termstrc, Journal of Statistical Software 36, No. 1, 1-34.

衍生品定价

- F. Black and M. Scholes (1973), The Pricing of Options and Corporate Liabilities, The Journal of Political Economy 81, No. 3, 637-654.
- J. Cox, S. Ross, and M. Rubinstein (1979), Option Pricing: A Simplified Approach, Journal of Financial Economics 7, No. 3, 229-263.
- D. Wuertz and many others (2012), fOptions: Basics of Option Valuation, R package version 2160.82. Available at <http://CRAN.R-project.org/package=fOptions>.
- R. C. Merton (1973), Theory of Rational Option Pricing, The Bell Journal of Economics and Management Science 4, No. 1, 141-183.
- R. Rebonato (1999), Volatility and Correlation, John Wiley, Chichester.
- S. Subramanian (2013), GUIDE: GUI for DERivatives in R, R package version 0.99.5. Available at <http://CRAN.R-project.org/package=GUIDE>.
- J. Hull (2011), Options, Futures, and Other Derivatives, Prentice Hall, 8th edition.

信用风险管理

- F. Black and J. Cox (1976), Valuing Corporate Securities: Some Effects of Bond Indenture Provisions, Journal of Finance 31, 351-367.
- D. Wuertz and many others (2012), fOptions: Basics of Option Valuation, R package version 2160.82. Available at <http://CRAN.R-project.org/package=fOptions>.

- K. Giesecke (2004), Credit Risk Modeling and Valuation: An Introduction. Available at SSRN: <http://ssrn.com/abstract=479323> or <http://dx.doi.org/10.2139/ssrn.479323>.
- I. Kojadinovic and J. Yan (2010), Modeling Multivariate Distributions with Continuous Margins Using the copula R Package, Journal of Statistical Software 34, No. 9, 1-20. Available at <http://www.jstatsoft.org/v34/i09>.
- J. Yan (2007), Enjoy the Joy of Copulas: With a Package Copula, Journal of Statistical Software 21, No. 4, 1-21. Available at <http://www.jstatsoft.org/v21/i04>.
- R. Merton (1974), On the Pricing of Corporate Debt: The Risk Structure of Interest Rates, Journal of Finance. 29, 449-470.
- D. Sharma (2011), Innovation in Corporate Credit Scoring: Z-Score Optimization. Available at SSRN: <http://ssrn.com/abstract=1963493> or <http://dx.doi.org/10.2139/ssrn.1963493>.
- S. M. Iacus (2009), sde: Simulation and Inference for Stochastic Differential Equations, R package version 2.0.10. Available at <http://CRAN.R-project.org/package=sde>.
- A. Wittmann (2007), CreditMetrics: Functions for calculating the CreditMetrics risk model, R package version 0.0-2.
- X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J. C. Sanchez, and M. Müller (2011), pROC: an open-source package for R and S+ to analyze and compare ROC curves, BMC Bioinformatics 12, 77.

极值理论

- E. Gilleland, M. Ribatet, and A. G. Stephenson (2013), A Software Review for Extreme Value Analysis, Extremes 16, 103-119.

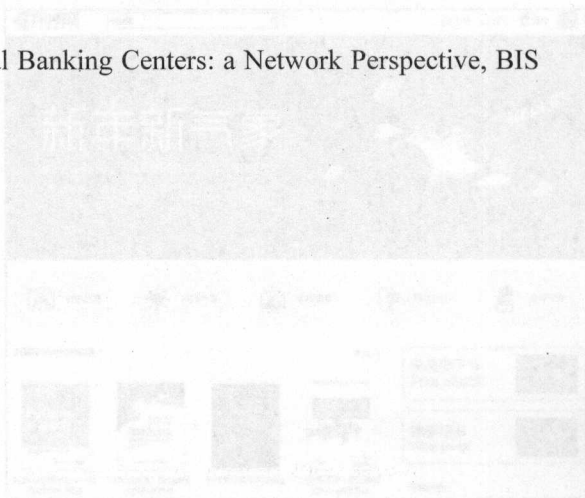
- A.J. McNeil, R. Frey, and P. Embrechts (2005), Quantitative Risk Management, Princeton University Press, Princeton.

金融网络

- A. L. Barabási and R. Albert (1999), Emergence of scaling in random networks, Science 286, 509-512.
- BCBS (2011), Global Systemically Important Banks: Assessment Methodology and the Additional Loss Absorbency Requirement, Committee on Banking Supervision. Available at <http://www.bis.org/publ/bcbs201.pdf>.
- M. L. Bech and E. Atalay (2008), The topology of the federal funds market, Federal Reserve Bank of New York, Staff Reports, 354.
- G. Bianconi and A. L. Barabási (2001), Competition and multiscaling in evolving networks, Europhysics Letters.54, 436.
- G. Csardi and T. Nepusz (2006), The igraph software package for complex network research, InterJournal, Complex Systems 1695. Available at <http://igraph.sf.net>.
- Z. Komárková, V. Hausenblas, and J. Frait (2012), How to Identify Systematically Important Financial Institutions, Report of the Central Bank of the Czech Republic, 100-111.
- Á. Lubláy (2006), Topology of the Hungarian large-value transfer system, MNB Occasional Papers, 57.
- S. Markose, S. Giansante, M. Gatkowski (2010), and A. R. Shaghghi, Too-Interconnected-To-Fail: Financial Contagion and Systemic Risk in Network Model of CDS and Other Credit Enhancement Obligations of U.S. Banks, COMISEF Working Paper Series, WPS-033-21-04-2010. Available at <http://comisef.eu/files/wps033.pdf> (downloaded on June 01, 2013).
- K. Soramäki, M. L. Bech, J. Arnold, R. J. Glass, and W.E Beyeler (2006), The

topology of interbank payment flows, Federal Reserve Bank of New York, Staff Reports, 243.

- G. von Peter (2007), International Banking Centers: a Network Perspective, BIS Quarterly Review.



社区里都有什么?

我们出版的图书涵盖主流IT技术, 在编程语言、Web 技术、数据科学等领域有众多经典畅销图书。社区现已上线图书1000余种, 电子书400多种, 部分新书实现纸书、电子书同步出版。我们还会定期发布新书书讯。

免费下载

社区内提供图书附赠的资源, 如书中的案例或程序源代码。

另外, 社区还提供了大量的免费电子书, 只要注册成为社区用户就可以免费下载。

与作者零距离

很多图书的作译者已经入驻社区, 您可以关注他们, 咨询技术问题, 可以阅读不断更新的技术文章, 听作译者和编辑聊聊好书背后的故事, 还可以参与社区的作者访谈栏目, 向您关注的作者提出采访题目。

灵活优惠的购书

您可以方便地下单购买纸质图书或电子书, 纸质图书直接从人民邮电出版社书库发货, 电子书提供多种阅读格式。

对于重磅新书, 社区提供预售和新书首发服务, 用户可以第一时间买到心仪的新书。

用户帐户中的积分可以用于购书优惠, 100积分=1元。选购图书时, 在“积分抵扣”处输入可使用的积分数值, 即可扣减相应金额。

欢迎来到异步社区！

异步社区的来历

异步社区 (www.epubit.com.cn) 是人民邮电出版社旗下 IT 专业图书旗舰社区，于 2015 年 8 月上线运营。

异步社区依托于人民邮电出版社 20 余年的 IT 专业优质出版资源和编辑策划团队，打造传统出版与电子出版和自出版结合、纸质书与电子书结合、传统印刷与 POD 按需印刷结合的出版平台，提供最新技术资讯，为作者和读者打造交流互动的平台。



社区里都有什么？

购买图书

我们出版的图书涵盖主流 IT 技术，在编程语言、Web 技术、数据科学等领域有众多经典畅销图书。社区现已上线图书 1000 余种，电子书 400 多种，部分新书实现纸书、电子书同步出版。我们还会定期发布新书书讯。

下载资源

社区内提供随书附赠的资源，如书中的案例或程序源代码。

另外，社区还提供了大量的免费电子书，只要注册成为社区用户就可以免费下载。

与作译者互动

很多图书的作译者已经入驻社区，您可以关注他们，咨询技术问题；可以阅读不断更新的技术文章，听作译者和编辑畅聊好书背后有趣的故事；还可以参与社区的作者访谈栏目，向您关注的作者提出采访题目。

灵活优惠的购书

您可以方便地下单购买纸质图书或电子图书，纸质图书直接从人民邮电出版社书库发货，电子书提供多种阅读格式。

对于重磅新书，社区提供预售和新书首发服务，用户可以第一时间买到心仪的新书。

用户帐户中的积分可以用于购书优惠。100 积分 = 1 元，购买图书时，在 里填入可使用的积分数值，即可扣减相应金额。

特别优惠

购买本书的读者专享异步社区购书优惠券。

使用方法：注册成为社区用户，在下单购书时输入 **S4XC5** **使用优惠码**，然后点击“使用优惠码”，即可在原折扣基础上享受全单9折优惠。（订单满39元即可使用，本优惠券只可使用一次）

纸电图书组合购买

社区独家提供纸质图书和电子书组合购买方式，价格优惠，一次购买，多种阅读选择。



社区里还可以做什么？

提交勘误

您可以在图书页面下方提交勘误，每条勘误被确认后可以获得 100 积分。热心勘误的读者还有机会参与书稿的审校和翻译工作。

写作

社区提供基于 Markdown 的写作环境，喜欢写作的您可以在这一试身手，在社区里分享您的技术心得和读书体会，更可以体验自出版的乐趣，轻松实现出版的梦想。

如果成为社区认证译者，还可以享受异步社区提供的作者专享特色服务。

会议活动早知道

您可以掌握 IT 圈的技术会议资讯，更有机会免费获赠大会门票。

加入异步

扫描任意二维码都能找到我们：



异步社区



微信服务号



微信订阅号



官方微博



QQ 群: 436746675

社区网址: www.epubit.com.cn

投稿 & 咨询: contact@epubit.com.cn

量化金融R语言 初级教程

本书将会教你如何利用统计计算语言R来解决现实世界的量化金融问题。本书包括了多种多样的主题，涉及从时间序列分析到金融网络的多个知识点。各章简要地给出了特定内容背后的理论，并且借助实际案例帮助读者使用R处理各种各样的问题。

本书在如何使用R来解决量化金融问题方面，是一本非常值得阅读的参考指南。本书包括了量化金融的必要内容，带读者了解R中许多清晰且实用的例子，不仅能帮助读者理解理论，而且能教会读者如何有效地处理自己现实生活中的问题。

本书以时间序列相关的内容开始，读者还将学到如何优化投资组合以及资产定价模型等内容。接下来，本书还包括了固定收益证券以及诸如信用风险管理等衍生产品。

本书的目标读者

如果你正在寻求使用R来解决量化金融问题的方法，本书将是你的理想之选。本书假定读者具有金融领域的基本知识，但并不要求读者熟悉R。本书致力于使用R来解决广泛的问题，为R的初学者和更有经验的用户提供了一系列有用的内容。



通过阅读本书，你将学到：

- 如何建模并预测房价、如何使用协整来改进对冲比以及如何建模波动率；
- 如何理解投资组合选择背后的理论以及如何把它运用到现实世界的的数据上；
- 如何利用资本资产定价模型以及套利定价理论；
- 如何理解固定收益产品的基础；
- 如何使用离散的和连续的时间模型来定价衍生证券；
- 如何成功地使用信用违约模型以及如何使用copula对相关的违约进行建模；
- 如何理解在保险和金融领域中极值理论的使用、模型的拟合以及风险度量的计算。



异步社区 www.epubit.com.cn
新浪微博 @人邮异步社区
投稿/反馈邮箱 contact@epubit.com.cn

美术编辑：董志桢

ISBN 978-7-115-45123-1



9 787115 451231 >

ISBN 978-7-115-45123-1

定价：49.00 元

分类建议：计算机 / 程序设计 / R

人民邮电出版社网址：www.ptpress.com.cn